Ph.D. DISSERTATION

Bayesian Prediction and Regression from Visual Data

영상 데이터에 대한 베이지안 예측 및 회귀 분석.

BY

YoungJoon Yoo

FEBRUARY 2017

DEPARTMENT OF ELECTRICAL ENGINEERING AND
COMPUTER SCIENCE
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Bayesian Prediction and Regression from Visual Data

영상 데이터에 대한 베이지안 예측 및 회귀 분석.

지도교수 최 진 영

이 논문을 공학박사 학위논문으로 제출함

2017 년 1 월

서울대학교 대학원

전기 컴퓨터 공학부

유 영 준

유 영 준의 공학박사 학위논문을 인준함

2017 년 2 월

위 원 장	조 남 익
부위원장	최 진 영
위 원	오 성 회
위 원	곽 노 준
위 원	최 상 일

Abstract

This dissertation proposes a new high dimensional regression / prediction method for diverse visual data pairs. In contrast to other regression / prediction methods, the proposed method focuses on the case where output responses are on a complex high dimensional manifold, such as images. In handling the complex data, the latent space embedding the information of the data is used for efficient regression / prediction. The dimensionality reduction methods into the latent space and the regression/prediction methods are designed as a Bayesian framework. For the prediction problem, the dissertation proposes a method to extract latent semantics on motion dynamics given in visual sequences. To this end, a Bayesian inference model is developed to capture the regional and temporal semantics of the dynamics data. The proposed Bayesian model is a hierarchical fusion of Gaussian mixture model and topic mixture model. It finds regional pattern information through topic mixture model and derives temporal co-occurrence of regional patterns through Gaussian mixture model. To infer the proposed model, the dissertation proposes a new sampling method that enables efficient inference. For the regression problem, we propose a method that makes a regression in the latent space for general and complex visual data pairs. This allows the latent space to imply the essential properties of the data pairs required for regression. For the purpose, a regression model is designed so that the regression in latent space should coincide with the regression in data space. The whole models are designed as Bayesian framework, and inferred by variational autoencoder framework.

Keywords: Regression, Prediction, Probablistic graphical model, Deep generative model, Approximate inference

Student Number: 2011-20884

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Finding an appropriate output response for the incoming input data with an unknown input/output relationship is one of the most crucial problems in data analysis. In diverse research fields, such as trajectory analysis, robotics, the stock market, etc. [1, 2, 2–4], target phenomena are interpreted as a form of paired input/output data. By finding the relationship between the input and output, unknown output responses for newly given input data can be inferred. This kind of problem is called as the regression, and we said the problem as the prediction in particular when estimating the future responses given the past data pair which composed of time series. The regression/prediction problems are theoretically well established and the analytic solutions for the infinite dimension of the basis function [5, 6] have been derived for the last century.

Many vision applications can also be expressed as such input/output data pairs. For example, as shown in Fig. 1.1 (a), Many visual sequences can be represented as data pairs defined as time-series. Also, as in Fig. 1.1 (b), the sequence of the motion images

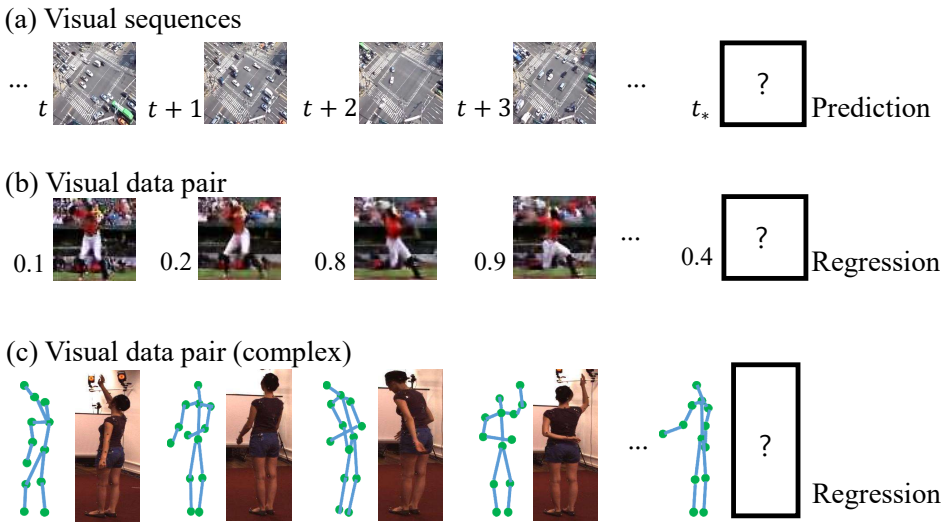


Figure 1.1: Examples of paired data in vision applications. (a) Given the visual sequences, the prediction problem can be defined. (b) Given the visual sequences, the regression problem can be also defined. In this case, the domain can be defined by the space representing relative orders of image sequences. (c) For joint-pose data pairs, the joint vector space can be a possible domain.

can be described by input/output paired data, where the input can be defined as the relative order and the output response is defined as the corresponding image. The motion capture data and their corresponding images in Fig. 1.1 (c) are another example. The input data are 3D joint positions, and their responses will be the corresponding posture images. If we can model the implicit function representing the given image data pairs via regression, we can estimate unobserved images that correspond to the input data. However, it is not straightforward to apply the existing multiple output regression algorithms [7–10] to these kinds of visual data applications, because the visual data are usually represented in compex and high dimensional spaces. In this case, projection of the data into lower dimensional space can be largely beneficial for alleviating the difficulty of the problem.

For the dimensionality reduction, a number of methods such as clustering [11, 12], eigen-vector analysis [13, 14], and latent variable method [15] are proposed. The kinds of methods can be described as a problem that infers the conditional relationship between data and the latent space. We call the approach as a generative model, and a Bayesian framework is one of the most representative method to design the generative model. In the Bayeisan fraemwork, a joint distribution regarding the latent space and the data space is defined, and the distribution is fitted to explain the observed data with the most highest probability. When defining the distribution for the latent space, the prior knowledge for the data can be used and advantegous to semi-supervised or weakly supervised data.

The objective of this dissertation is to develop a regression/prediction method handling the meaningful semantics in the latent space designed in Bayesian perspective. For the prediction problem, we deal with a prediction problem that estimates the future position of an object in a visual sequence. For the regression problem, we propose a Bayesian regression algorithm to estimate an unknown output scene for a given input knowledge in visual input / output pairs.. The target method aims to effectively predict

/ regress high-dimensional visual data by reducing the dimension of data by applying a Bayeisan generative method, and to make the learned latent space compress the core characteristics of data.

1.2 Related Work

Regression / prediction of paired data is theoretically well established and analytic solutions for the infinite dimension of the basis function [5, 6, 16] have been derived for the last century. In non-parametric cases, Gaussian process [6, 15] provides a general solution by expanding Bayesian linear regression using kernel metrics and a Gaussian process prior. By using this method, we can estimate the output data as a Gaussian posterior composed of given data pairs and input data to the unobserved target outputs. However, applying the algorithms for the high-dimensional output data is difficult because the kernel metric has limited capacity to express the complicated high dimensional data. The variants of multiple output regression algorithms [7–10] are proposed to deal with multi-dimensional output responses. Still, these algorithms focus on handling relatively low dimensional output responses and are not able to sufficiently describe complicated data, such as that of an image. Therefore, the latent space capturing the essential semantics of the data should be combined to the prediction/regression model to successful estimation of the output responses.

For the case when the data is to model the movement patterns in a visual sequence, various pattern analysis algorithms based on the probabilistic topic model [17–24] are proposed to learn object dynamics in a scene. The existing topic model-based algorithms learns the regional patterns in a form appropriate to judge whether or not the target is moving in the typical regions. To progress one step further to those algorithms, the object moving dynamics should be learned in the form of moving patterns together with their co-occurrences in a way that is adequate for prediction of future path.

The recent path prediction research can be categorized into two approaches: path-planning-based approach and patch-appearance-based approach. The first approach utilizes a path planning algorithm [25–29]. The approach uses statistical techniques such as inverse reinforcement learning [30–32] to find the optimal future path. Kitani *et al.* [33] first utilized the robot path planning algorithm to infer a point-wise future location of an object in a visual scene. The goal of this algorithm is to find a well-planned path for a target object with given scene structures such as roads, buildings, and so on. The object passes the appropriate area such as the pavement or road and avoids static obstacles in its way by following the induced path to reach the destination. To infer the predicted path, the algorithm first finds the cost for accessing each location in a scene and describes the cost via the reward map by utilizing the semantic segmentation result [34]. Then the algorithm extracts the optimal path which minimizes the overall cost by using inverse optimal control [30] and Markov decision process [35]. This approach is designed for single object movement prediction and does not consider possible collisions with other moving objects in a scene.

The second approach induces future changes of notable patches instead of locations of the target. In this case, inferring the representative patches is also a subproblem to be solved. Walker *et al.* [36] found the salient patches by applying recent mid-level patch-finding algorithms [37–41]. Then, they generated the weighted graph explaining the changes of the patches. The nodes of the graph represent the future locations and shapes of the patch. The weight is defined as a transition cost. The algorithm then finds the minimal weighted path by using Dijkstra's algorithm [42]. This path, starting from the initial node to termination, describes the changes of both shape and location. However, this algorithm has also been designed for single patches and does not reflect the dynamics of other moving objects in the scene. Unlike the existing approaches, we will present a path prediction algorithm that reflects the movements of other co-occurring objects.

Regarding the regression problem, the probabilistic generative models [11, 43–48] applied to be formal problem have proven to be successful in understanding diverse unsupervised data, but their descriptive ability is insufficient to fully explain complex data such as images [49]. Recently, as in other works in the vision area [50, 51], deep layered architectures have been successfully applied to solve this problem with powerful data generation performance. Among these architectures, generative adversarial network (GAN) [52] and generative moment matching networks (GMMN) [53] directly learn the generator that maps latent space to data space. Meanwhile, the variants of the restricted Boltzman machine (RBM) [49, 54–56] and probabilistic autoencoders [57–59] learn the encoder that defines the map from data to latent space and the generator (decoder) simultaneously. The former methods, and especially variants of GAN [52, 60, 61], are reported to describe the edges of generated images more sharply than the latter methods. However, the applicability of these methods is restricted due to the difficulty of discovering the relationships between data and latent space. This innate nature makes it difficult to use adversarial networks for designing the regression. Therefore, this paper adopts the variational autoencoder framework [57], which is also more suitable than RBM families to expand the regression model.

Since Kingma *et al* [57] first published the variational autoencoder (VAE), numerous applications [62, 63] have been presented to solve various problems. Yan *et al* [63] proposed conditional VAE in order to generate the image conditioned on the attribute given in the form of sentences. Furthermore, recent work [64–66] has demonstrated that a sequence in latent space would be mapped back to the sequence of data. Hence these methods embedded dynamic models such as recurrent neural networks [64] and the Kalman filter [16, 65] into the VAE framework. These algorithms [64, 65, 65, 66] successfully show the ability of dynamic models in a latent space to capture the temporal changes of relatively simple objects in images. In this paper, we apply the VAE for the regression task in a relatively complex manifold.

1.3 Contents of Research

In this dissertation, a latent space is designed to embed the essential semantics of observed data, and a Bayesian method is developed to infer the proper output responses for given new input data by utilizing the latent space. First, we propose a prediction model that can handle dynamic semantics from the visual sequences, then we propose a regression model that estimates unobserved images from the observed visual data with input (clue) information. For the prediction case, we focus on handling the object moving semantics in a crowded scene. These semantics include diverse movement patterns and the spatio-temporal relationship among the patterns, according to the scene structure. We design a new probabilistic method capturing the characteristics of the movement patterns and propose a movement prediction algorithm generating the future movement of the objects in a scene. In particular, we develop a new unsupervised Bayesian learning model that extracts typical movement patterns of objects and relationships among the patterns to solve the prediction problem. The proposed model combines a topic mixture model [67] and the Gaussian mixture [68] hierarchically, which learns movement patterns as well as their interactions by utilizing the feature tracking results. However, the hierarchical combination of these two mixture models is not mathematically straightforward because the Gaussian distribution is not a conjugate prior [69] of multinomial (topic) distribution, and so the posterior distribution of the combined model cannot be derived. Hence, this kind of combination has not been utilized despite its effectiveness. To resolve the problem, we introduce a mathematical trick to formulate a hierarchical topic-Gaussian mixture with satisfying the conjugate prior relation through an augmented variable. Then, we develop a deterministic path prediction algorithm utilizing the moving dynamics inferred by the proposed hierarchical topic-Gaussian mixture model. In this algorithm, we predict the future path of the target object by inferring the most plausible movement pattern for the ob-

ject through analysis of the previous location of the object and moving dynamics of other co-occurring objects.

For the regression case, we propose a regression method to handle general and complex input/output data pair. Specifically, we handle the case when the output response is visual data such as images. For dealing with the case, we solve this problem by combining the VAE [57] and Gaussian process regression [6]. The key idea of this work is to do regression in the latent space instead of the high-dimensional image space. The proposed algorithm generates the latent space that compresses the information of both the domain and output image using the VAE, and projects the data pairs to the latent space. Then, regression is conducted for the projected data pairs in latent space, and the decoder is trained so that the regression in latent space and image space coincide. The whole process, including the loss function, is designed as the generative model, and a new mini-batch composition method is presented for training the decoder to satisfy our purpose. All connection parameters of the encoder / decoder are inferred by the end-to-end stochastic gradient descent method as described in [70]. The proposed regression method is validated with two different examples: sports sequences and motion image sequences with skeletons. The first example presents a regression case of simple domain to complex codomain, and the second example presents the complex domain to complex codomain case.

1.4 Thesis Organization

In Chapter 2, as for the preliminaries, we briefly discuss about topics in Bayesian framework. First, the definition and configuration of the Bayesian model are introduced. Second, the introduction of approximate inference methods for the Bayesian model is presented. Third, recent variational autoencder frameworks explaining the general Bayesian inference as the autoencoder framework, are explained. Last, we re-

view Gaussian process regression which is a general nonparameteric regression method given Gaussian prior. Chapter 3 describes the prediction method of object movements in image sequences such as traffic scenes. Detailed explanation of the model configuration and the inference method is presented. Chapter 4 presents a variational autoencoded regression method which can handling diverse visual data pairs. We present the detailed explanation of the model configuration and the inference method. In Chapter 5, the qualitative/quantitative results for the proposed methods are presented. The experimental results of the prediction method are introduced with the various traffic scenes in section 5.1. In section 5.2, The proposed regression method is tested with two cases: sports sequences and human-joint pair. The analysis and discussion for the results from the two cases are presented. In Chapter 6, we conclude by summarizing the contributions of our work and briefly mentioning the direction of our future research.

Chapter 2

Preliminaries

2.1 Overview

In this chapter, we introduce the main theories used in this paper. First, in chapter 2.2, we describe Bayesian statistics and the relationship between the statistics and generative models. Next, Chapters 2.3 and 2.4 introduce the inference of the Bayesian model. Chapter 2.3 describes the sampling-based method and Chapter 2.4 describes the optimization-based method. Also, regression methods including Gaussian process regression are described in Chapter 2.5. For more detailed explanation, refer to the cited literatures.

2.2 Bayesian Statistics for Generative Model

2.2.1 Overview

The model that can sample the new data which mimicking the semantics of the observed data is called generative model. In statistics view, This model can be explained

as a procedure of sampling the data x_* from the data distribution $p(x)$ obtained from the given data set $X = \{x\}$. Typically, we define a latent variable z to describe the semantics of the data X to define the distribution $p(x), x \in X$.

The direction of the solution changes depending on whether (1) z is assumed to be an unknown fixed value or (2) z is defined as random variable. In case (1), z is the unknown fixed value, $x \in X$ is the value sampled by z, and $p(x|z)$ is the freqeuncy of the sampled x. In this case, z maximizing the probability $p(x|z)$ for the given sample x is estimated by a deterministic method. The above view of latent variable is called as the frequentist view, which defines the probability as a long run frequencies of samples from an event [71]. Conversly, in case (2), z is defined as random variable with $p(z)$. In this case, $p(z)$ does not fit the definition of the probability that the frequentist thinks. $p(z)$ is not the frequencies of observed samples, but denotes a measure quantifying uncertainty about z [72]. Then, the latent variable z is realized by the $p(z|x)$ when the data x is given. This view of latent variable is called as Bayesian statistics view that we are interested in. The Bayesian approach provides some practical approaches of exploring the latent space such as sampling and it is advantageous in solving the generative model problem. A detailed description of Bayesian statistics is provided below.

2.2.2 Baye's Theorem

Mathematically, we infer the posterior distribution $p(z|x)$, which encapsulates everything we know about the unknown z, by equation (2.1),

$$p(z|x) = \frac{p(x,z)}{p(x)}. \tag{2.1}$$

The joint distribution $p(x,z) = p(x|z)p(z)$ is defined by the **likelihood function** $p(x|z)$ and the **prior** $p(z)$. The **prior** probability $p(z)$ captures our assumption about z, before observing the data. The **likelihood** is evaluated for the observed data X and

can be viewed as a function of z. It expresses how probable the observed data set is for different setting of the latent variable z. We note that the likelihood is not a probability distribution over z, and hence its integral with respect to z deos not neccessarily equal to one. The **denominator** $p(x)$ is the normalization constant, which ensures that the posterior probability $p(z|x)$ is a valid. It is important that the distribution $p(x)$ is function of observed data $x \in X$ and does not consider newly sampled data x_*. In most case, $p(x|z)$ and $p(z)$ are design factors and we can exactly find the posterior $p(z|x)$ by marginalizing the joint pdf $p(x, z)$. We introduce an example by solving a Bayesian curve fitting problem in the following.

2.2.3 Example: Bayesian Curve Fitting Problem

To show the example for the Bayesian statistics, we introduce the Bayeian regression example, described in [5] in detail. Let us assume we have N number of input values $\mathbf{x} = [x_1, ..., x_N]^T$ and their corresponding response $\mathbf{r} = [r_1, r_2, ..., r_N]^T$. We shall assume that, given the value of x, the corresponding r has a Gaussian distribution with a mean equal to the value $y(x, \mathbf{z}) = \sum_{j=1}^{M} z_j x^j, \mathbf{z} = [z_1, ..., z_M]^T$. Thus, the response r defined as

$$p(r|x, z, \rho) = \mathcal{N}(y(x, \mathbf{z}), \rho^{-1}), \tag{2.2}$$

where the precision parameter ρ is defined as a inverse variance of the distribution.

Next, we determine the values of the unknown parameters $\mathbf{z}$ and ρ by using the training data $\mathbf{x}$ and $\mathbf{r}$. If the data are independent and identically distributed (iid), then the log-likelihood function is given by

$$\ln p(\mathbf{r}|\mathbf{x}, \mathbf{z}, \rho) = -\frac{\rho}{2} \sum_{n=1}^{N} \{y(x_n, \mathbf{z}) - r_n\}^2 + \frac{N}{2} \ln \rho - \frac{N}{2} \ln 2\pi \tag{2.3}$$

Consider first the determination of $\mathbf{z}$ by frequentist approach. Then $\mathbf{z}$ is estimated by

maximizing (2.3) with respect to $\mathbf{z}$, which will be denoted by $\mathbf{z}_{ML}$. Since the last two terms are not dependent on $\mathbf{z}$, we can ommit the terms and only the first term is considered. Also, we can replace ρ to 1 for estimating $\mathbf{z}$ because a scale of the positive constant coefficient does not changes the location of the maximum regarding $\mathbf{z}$. Therfore, in frequentist view, maximizing the likelihood with respect to $\mathbf{z}$ is equivalent to minimizing the **sum-of-squared** error. Through the same process, we can find the value of ρ, which is

$$\frac{1}{\rho_{ML}} = \frac{1}{N}\sum_{n=1}^{N}\{y(x_n, \mathbf{z}_{ML}) - r_n\}^2. \tag{2.4}$$

One step toward to Bayesian, we can impose a prior distribution $p(\mathbf{z})$ over $\mathbf{z}$. Let assume a multi-variate zero mean Gaussian prior distribution of the form

$$p(\mathbf{z}|\alpha) = \mathcal{N}(\mathbf{z}|\mathbf{0}, \alpha^{-1}I), \tag{2.5}$$

where α is the **hyperparameter** of $\mathbf{z}$. Then, according to Baye's theorem, the posterior distribution for $\mathbf{z}$ is proportional to the product of the likelihood and the prior

$$p(\mathbf{z}|\mathbf{x}, \mathbf{r}, \alpha, \rho) \propto p(\mathbf{r}|\mathbf{x}, \mathbf{z}, \alpha, \rho)p(\mathbf{z}|\alpha). \tag{2.6}$$

By taking the negative logarism of (2.6) and combining with (2.5) and (2.3), we can induce that the maximum the posterior is equivalent to the mimimum of

$$\frac{\rho}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{z}) - r_n\}^2 + \frac{\alpha}{2}\mathbf{z}^T\mathbf{z}, \tag{2.7}$$

which is same as a two-norm regularized **sum-of-squared** error. We note that this is not a Bayesian even though we have included the prior distribution. It is because we still make a point-estimate of $\mathbf{z}$. In Bayesian approach, we should infer the posterior distribution of $\mathbf{z}$ and it is possible by marginalizing the joint pdf.

In the curve fitting problem, given the training data $\mathbf{x}$ and $\mathbf{r}$, our goal is to predict the value of r_* for a new data point x_*. We therefore wish to evaluate the predictive

distribution $p(r_*|x_*, \mathbf{x}, \mathbf{r})$. For simplicity, we assume that the parameters α and ρ are fixed in advance.

A Bayesian treatment simply corresponds to a consistent application of the sum and product rules of probability, which allows the predictive distribution to be written in the form

$$p(r_*|x_*, \mathbf{x}, \mathbf{r}) = \int p(r_*|x_*, \mathbf{z})p(\mathbf{z}|\mathbf{x}, \mathbf{r})d\mathbf{z}, \tag{2.8}$$

where $p(r_*|x_*, \mathbf{z})$ is given by equation (2.3) where $p(\mathbf{z}|\mathbf{x}, \mathbf{r})$ is the posterior distribution over parameters, and can be found by normalizing the right-hand side of equation (2.6), which is the joint pdf over $\mathbf{z}, \mathbf{x}$ and $\mathbf{r}$. For this example, the overall calculation of integration (2.8) is solved analytically. The resultant predictive distribution (2.8) is given by

$$p(r_*|x_*, \mathbf{x}, \mathbf{r}) = \mathcal{N}(r_*|m(x_*), s^2(x_*)), \tag{2.9}$$

where

$$m(x_*) = \rho\phi(x_*)^T H \sum_{n=1}^{N} \phi(x_n)r_n, \tag{2.10}$$

$$s^2(x_*) = \rho^{-1} + \phi(x_*)^T H \phi(x_*). \tag{2.11}$$

The matrix H is defined by

$$H = (\alpha I + \rho \sum_{n=1}^{N} \phi(x_n)\phi(x_*)^T), \tag{2.12}$$

where $\phi(x) = [x\ x^2\ ...\ x^M]^T$. We can see that in equation (2.12), the second term is derived from the Bayesian approach to handle the uncertainty of the $\mathbf{z}$. For more detailed explanation, refer to [73].

2.2.4 Bayesian Model Comparison

To solve the curve fitting problem, we illustrate mainly two different perspectives; frequentist's view and Bayesian's view. In frequentist view, the point-estimate of the

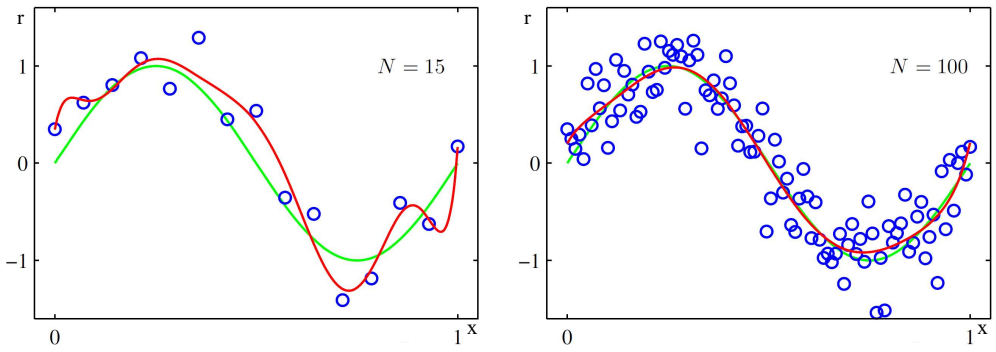


Figure 2.1: Overfitting example from [73]. polynomial line fitting example by minimizing sum-of-squared error in equation (2.2). $M = 9$ degree polynomial is used for $N = 15$ and 100 samples.

parameter **z** is conducted by maximizing the likelihood function. In this approach, selecting the complexity of the model is very important because of so called **over-fitting** problem. As seen in the figure 2.1, If the size of the model is too large compared to the number of learning data, a problem arises from that the model is excessively fitted for the learning data. In our polynomial fitting problem, the complexity of the model is governed by the degree of polynomial of $\phi(x)$ and we will call the model as probability distribution (2.8) over observation **x**.

In Bayesian's perspective, this over-fitting problem is proven to be less severe by the Bayesian model comparison. Suppose we wish to compare a set of L models m_i where $i = 1 \ldots L$. We assume that the response r_* is inferred from one of L models. Now, our goal is to find the most probable m_i for data **x** by seeing $p(m_i|\mathbf{x})$, which is proportional to the joint distribution. Here the posterior distribution $p(m_i|\mathbf{x})$ is given in

$$p(m_i|\mathbf{x}) = \frac{p(m_i)p(\mathbf{x}|m_i)}{p(\mathbf{x})}. \tag{2.13}$$

For analyzing m_i, the denominator $p(\mathbf{x})$ is negligible because it is constant with respect

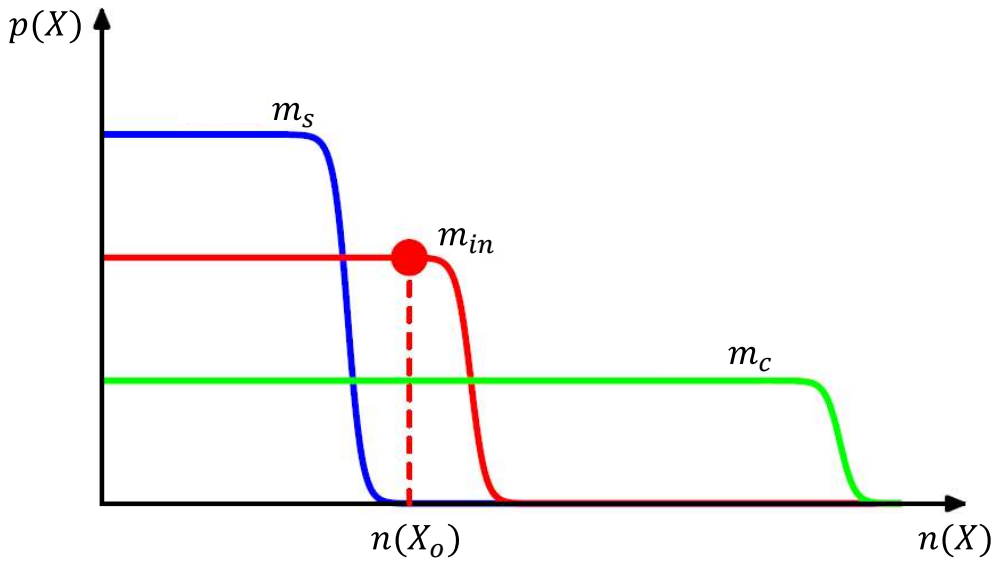


Figure 2.2: Schematic illustration of the number of data and the complexity of the model from [73]. M_s is the most simplest model and M_{in} is the intermediate model. M_c is the most complex model among them. It is important that the complex model does not always have the highest evidence for all dataset.

to m_i. Also, if we assume $p(m_i)$ to be equal for all the models, the only terms governing the posterior is the likelihood $p(\mathbf{x}|m_i)$. The likelihood term $p(\mathbf{x}|m_i)$ so called **model evidence** $p(\mathbf{x}|m_i)$ and it refers to the preference of the data for each model. In our curve-fitting example, the model m_i is governed by $\mathbf{z}$ and the model evidence term is given by

$$p(\mathbf{x}|m_i) = \int p(\mathbf{x}|\mathbf{z}, m_i)p(\mathbf{z}|m_i)d\mathbf{z}. \tag{2.14}$$

Then, let the complex model as m_c and the simple model as m_s. In most case, the complex model has higher likelihood; it means $p(\mathbf{x}|\mathbf{z}, m_c) > p(\mathbf{x}|\mathbf{z}, m_s)$ in general case. However, for complex model, the solution space of the latent variable $\mathbf{z}|m_c$ is much larger than simpler variable $\mathbf{z}|m_s$. Since the prior $p(\mathbf{z}|m_i)$ should be equal to one, this makes it difficult for the model to allocate sufficient probability density for the most suitable parameter for the data. Therefore, if both models have similar likelihoods for the data, then the simpler model m_s has a higher probability of model selection. See figure 2.2. This characteristic is also called Occam's razor.

2.2.5 Approximate Inference

So far, we have explained the concept of the Bayesian treatment through the curve-fitting example. In the example, the marginalization of the model is analytically tractable. However, the inference of posterior is intractable in most case, because the calculation of the evidence term $p(x) = \int p(x, z)dz$ is not possible in most case. Therefore, approximate inference methods are required to infer $p(z|x)$.

The approximation methods are mainly categorized into two methods. One is a sampling based approach using Markov chain Monte carlo (MCMC) and the other is optimization based method using the variational inference. The former is a method of matching the z obtained by the iterative sampling from a proposal distribution with the sample from the true posterior. This approach is proven to access to the true posterior, but it is difficult to handle large data since it is hard to apply the stochastic approxima-

tion [74]. The latter is a method of minimizing the KL-divergence between $q(z)$ and the posterior $p(z|x)$, where $q(z)$ is the distribution approximating $p(z|x)$. Although $q(z)$ is not fully accessible to true posterior $p(z|x)$, It is widely used for handling large amount of data since it is an optimization problem that stochastic approximation can be applied [74]. In the following section, we describe a detailed description of each method.

2.3 Sampling Based Methods

2.3.1 Overview

The Monte carlo approximation through sampling is a typical method when it is difficult to calculate the integration of probability distribution. The method is proven to converge to the solution when the number of sample is infinity, but it is impractical. Therefore, diverse efficient sampling approches are presented. Among them, we will mainly discuss about the widely used Markov chain Monte Carlo (MCMC) method. MCMC method first defines a chain of iterative samples and then, proposes requirements for samples to converge to that of target distribution. The following section describes the detailed formulation and convergence requirements for the Markov Chain Monte Carlo method.

2.3.2 Monte Carlo Method

In most situations, the posterior distribution is required for the purpose of obtaining expectations, for example in order to make predictions. So the fundamental problem we are trying to solve is finding expectations for some function $f(z)$ with respect to a probability distribution $p(z)$. In case of continuous variable z, our goal is to solve the expectation

$$E[f] = \int f(z)p(z)dz. \tag{2.15}$$

If z is discrete variable, the integral will be substituted to a summation. In many case, the expectations are too complex to be empirically calculated using analytical methods. The key idea of the Monte Carlo approximation is to get independent samples $z^{(i)}$ from the distribution $p(z)$ and use it for approximation. Using the set of sample $z^{(i)}, i = 1 \dots N$, we can approximate the expectation in equation (2.15) by

$$\bar{f} = \frac{1}{N} \sum_{i=1}^{N} f(z^{(i)}). \tag{2.16}$$

We can easily conform that the estimator $\bar{f}$ has correct mean because $E[\bar{f}] = E[f]$. According to the 'law of large numbers', the estimator converges to real expectatation and practically, we can expect high accuracy when given about ten to twenty independent samples [73]. However, the samples $z^{(i)}$ are not usually independent and drawing samples from the distribution $p(z)$ is not a trivial problem. In the section below, we introduce some sampling methods to solve these difficulties.

2.3.3 Basic Sampling Methods

For standard non-uniform distributions, it is possible to draw the samples from uniformly distributed random numbers. Let us assume that z is drawn from uniform distribution with interval $(0, 1)$. Our goal is to transform the value of z into y which is the samples from $p(z)$. Then the distribution of y is governed by

$$p(y) = p(z)|\frac{dz}{dy}|, \tag{2.17}$$

where $p(z) = 1$ in this case. Therefore, we obtain

$$z = c(y) = \int_{-\inf}^{y} p(\hat{y}) d\hat{y}, \tag{2.18}$$

by integrating (2.17). Thus, $y = c^{-1}(z)$ and we can transform uniform random samples to target distribution. However, in most case, the indefinite integral (2.18) is analytically intractable. Therefore, we introduce alternative sampling approaches in below section.

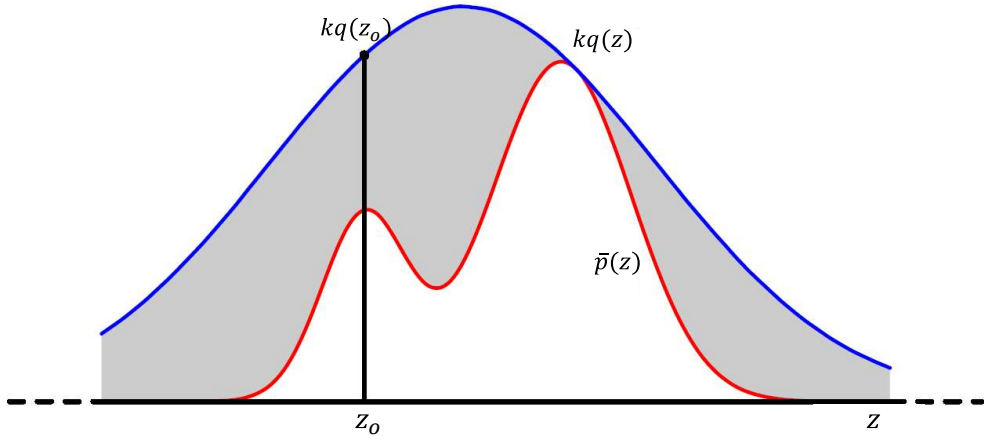


Figure 2.3: The illustration describing the rejection sampling [73]. The samples fall in gray area are rejected.

2.3.3.1 Rejection Sampling

The rejection sampling enables us to draw samples from relatively complex distributions. Suppose that our target distribution $p(z)$ is a relatively complex distribution that is hard to evaluate the indefinite integration as in (2.18). Then it is difficult to apply the standard sampling techniques from uniform distribution. Also, let us assume that we can easily evaluate $p(z)$ for any given z, up to some given denominator Z_p. It means

$$p(z) = \frac{\bar{p}(z)}{Z_p}, \tag{2.19}$$

where $\bar{p}(z)$ can be evaluated and Z_p is unknown. To draw samples from $p(z)$, we use more simpler deistribution $q(z)$ that we can readily draw samples. The distribution $q(z)$ is called **proposal distribution**.

Next, we define a constant k such that $kq(z) \geq p(z)$ for all values of z. The function $kq(z)$ is called **comparison function**. Then, the steps for rejection sampling is as follows. First, we sample z_o from $q(z)$. Second, we generate a sample u_o from the uniform distribution over interval $[0, kq(z_o)]$. Finally, if the sample u_o is smaller than $\bar{p}(z_o)$, the sample u_o is retained and otherwise rejected. We note that the remaining

samples are uniformly distributed under the curve of $\bar{p}(z)$ as shown in figure 2.3, and hence the samples are distributed according to $\bar{p}(z)$. The acceptance ratio $p(accept)$ is given by

$$p(accept) = \int \frac{\bar{p}(z)}{kq(z)} q(z)dz = \quad \frac{1}{k} \int \bar{p}(z)dz. \tag{2.20}$$

Therefore the selection of k is crucial for efficient sampling. Unfortunately, the selection of k is not trivial in many case because it is difficult to analytically solve the equation $\bar{p}(z) = kq(z)$ for most $\bar{p}(z)$.

2.3.3.2 Importance Sampling

In many case, the desirable method of probability distribution analysis is to evaluate expectation, not the posterior distribution itself. The **importance sampling** approach provides a method for approximating the expectation directly but does not provide a way to draw samples from $p(z)$.

The Monte Carlo approximation of the expectation is given by (2.16). In many case, it is difficult to draw samples directly from the distribution $p(z)$ but it is possible to evaluate $p(z)$ for any given z. Therefore, if we descretize z-space into a uniform grid, then we can approximate the expectation by

$$E[f] \simeq \sum_{l=1}^{L} p(z^{(l)}) f(z^{(l)}). \tag{2.21}$$

An obvious problem with this approach is that the amount of computation increases exponentially with the dimension of z. Moreover, uniform sampling can be particularly inefficient for high dimensional problems, since many types of probability distributions often have masses confined to small areas of $z-$space. Our goal is to select as many sample points as possible in a region where $p(z)$ is large, or ideally $p(z)f(z)$ is large.

Same as rejection sampling, Importance sampling also uses proposal distribution $q(z)$ to draw samples. Then we can express the equation (2.21) over the samples $\{z^{(l)}\}$ drawn from $q(z)$

$$\begin{aligned} E[f] &= \int f(z)p(z)dz \\ &= \int f(z)\frac{p(z)}{q(z)}q(z)dz \\ &\simeq \frac{1}{L}\sum_{l=1}^{L}\frac{p(z^{(l)})}{q(z^{(l)})}f(z^{(l)}). \end{aligned} \tag{2.22}$$

The quantity $p(z^{(l)})/q(z^{(l)})$ are called as **importance weight**. Unlike rejection sampling, no sample is rejected in importance sampling.

When the case the $\bar{p}(z)$ of $p(z) = \bar{p}(z)/Z_p$ is accessible and the normalization constant Z_p is unknown, we can use importance sampling by introducing the proposal distribution $q(z) = \bar{q}(z)/Z_q$ which has same property to the $p(z)$. Then the expectation is derived as

$$\begin{aligned} E[f] &= \int f(z)p(z)dz \\ &= \frac{Z_q}{Z_p}\int f(z)\frac{\bar{p}(z)}{\bar{q}(z)}q(z)dz \\ &\simeq \frac{Z_q}{Z_p}\frac{1}{L}\sum_{l=1}^{L}\frac{\bar{p}(z^{(l)})}{\bar{q}(z^{(l)})}f(z^{(l)}). \end{aligned} \tag{2.23}$$

The unknown constant Z_q/Z_p can be evaluated as follow,

$$\begin{aligned} \frac{Z_p}{Z_q} &= \frac{1}{Z_q}\int \bar{p(z)}dz \\ &= \int \frac{\bar{p}(z)}{\bar{q}(z)}q(z)dz \\ &\simeq \frac{1}{L}\sum_{l=1}^{L}\frac{\bar{p}(z^{(l)})}{\bar{q}(z^{(l)})}, \end{aligned} \tag{2.24}$$

and hence

$$E[f] \simeq \sum_{l=1}^{L} w_l f(z^{(l)}),\ w_l = \frac{\bar{p}(z^{(l)})/\bar{q}(z^{(l)})}{\sum_m \bar{p}(z^{(m)})/\bar{q}(z^{(m)})}. \tag{2.25}$$

As with rejection sampling, the success of the importance sampling method is determined critically by how well the sampling distribution $q(z)$ matches the desired distribution $p(z)$. If $p(z)f(z)$ changes strongly and a significant portion of its mass is confined in a relatively small region of z space, the importance weight can be dominated by a small number of weights which have large values. Correspondingly, the effective sample size may be much smaller than the given sample size L. In addition, the problem becomes more serious when there is no sample in a region where $p(z)f(z)$ is large. This highlights the key requirements for the sampling distribution $q(z)$. That is, the number of samples in areas where $p(z)$ is important should not be small or zero.

2.3.4 Markov Chain Monte Carlo

In the previous section, we discussed rejection sampling and importance sampling strategies to evaluate the expectations of a function, and found that it suffered from serious limitations, especially in high-dimensional spaces. Therefore, we use a generic and powerful framework called the **Markov chain Monte Carlo (MCMC)** which allows sampling from large class of distributions and can manage the large dimensions of the sample space.

The Markov Chain Monte Carlo (MCMC) is an algorithm in which the sample z_i of the state Z is connected to the markov chain and iteratively discovers the state space. The purpose of the algorithm is to let the sampled $z^{(i)}$ mimick the samples from the posterior $p(z|X)$, where X is the set of data x.

For simplicity, $z^{(i)}$ is assume to be in descrete space in this section, where $z^{(i)} \in \{z_1, z_2, ..., z_s\}$. The stochastic process z^i is called as **Markov process** when satisfy the equation (2.26).

$$p(z^{(i)}|z^{(1)}, z^{(2)}, \cdots, z^{(i-1)}) = p(z^{(i)}|z^{(i-1)}). \tag{2.26}$$

This chain converges to the invariant distribution $p(z)$ when satisfying the two condi-

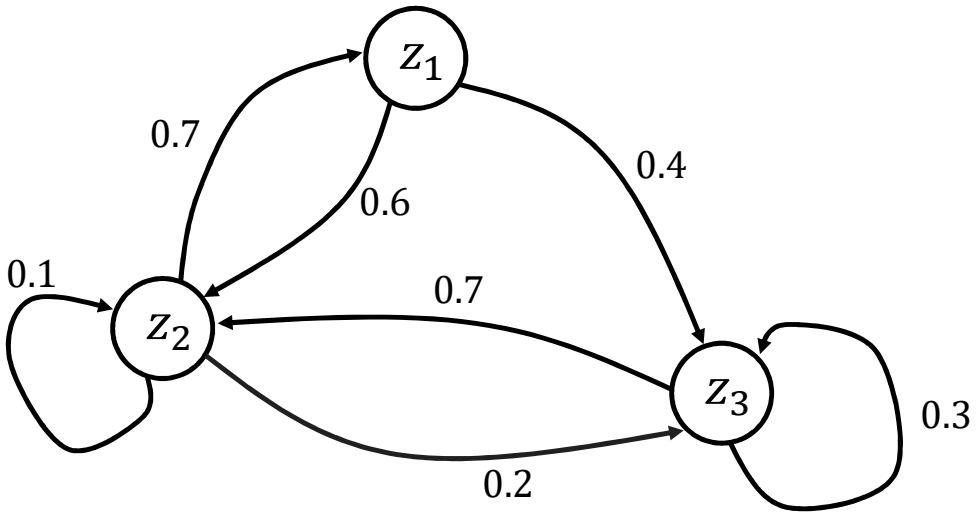


Figure 2.4: The example of Markov chain. In this example, we assume 3 states and the values of each direction indicate the transition probability between states.

tion below.

(1) **Irreducibility**: For any state of the Markov chain, there is a positive probability of visiting all other states.

(2) **Aperiodicity**: The chain should not get trapped in cycles.

To satisfy both conditions, we introduce the detailed balance (reversibility) condition in equation (2.27), which is the sufficient condition for the two requirements.

$$p(z^{(i)})p(z^{(i)}|z^{(i-1)}) = p(z^{(i-1)})p(z^{(i-1)}|z^{(i)}). \tag{2.27}$$

In MCMC algorithm, Markov chain should be irreducible and aperiodic, and these can be satisfied when the chain satisfies the detailed balance condition of (2.27). Metropolis hastings [75] algorirthm is the most popular algorithm which is designed to satisfy the detailed balance condition. Let us assume that z_c and z_f be the past sample and newly extracted sample respectively. Metropolis Hastings (MH) algorithm uses a proposal $q(z_f|z_c)$ to access the $p(z)$, and acceptance probability $A(z_f, z_c)$ to decide

whether to accecpt the sample z_f or not. The proposal $q(.)$ represents the transition probability, and acceptance probability is used to satisfy the detailed balance condition. Futhermore, for arbitarary $q(.)$, the MH chain is proven to satsify the detailed balance condition when the acceptance probability is defined as equation (2.28),

$$A(z_f, z_c) = min\{1, \frac{p(z_f)q(z_c|z_f)}{p(z_c)q(z_f|z_c)}\}. \tag{2.28}$$

To use this algorithms in practice, it is important to well define the proposal distribution. If the proposal distribution largely makes samples that not accepted, the convergence speed will be slow. Hence, we introduce the Gibbs sampler which is one of the most popular algorithm handling the problem.

2.3.5 Gibbs Sampling

Gibbs sampler is a kind of MH sampler that updates the sample through a data driven proposal. Suppose $z = [z_1, z_2, ..., z_N]$ is N dimensional vector, and assume the conditional expression $p(z_j|z_{-j}), z_{-j} == \{z_1, z_2, \cdots, z_{j-1}, z_{j+1}, ..., z_N\}$. In gibbs sampler, we iteratively draw a new sample for each element z_j of z by using the samples z_{-j} in past iteration. Then, the ith sample for z_j is defined by using the conditional distribution $p(z_j|z_{-j}^{i-1})$, as in

$$q(z|z^{(i-1)}) = \begin{cases} p(z_j|z_{-j}^{(i-1)}),\ if\ z_{-j} = z_{-j}^{(i-1)} \\ 0, \end{cases}, \tag{2.29}$$

where $z_{-j}^{(i)} = \{z_1^{(i)}, ..., z_{j-1}^{(i)}, z_{j+1}^{(i)}, ..., z_N^{(i)}\}$. Then, combining (2.29) to the acceptance ratio in (2.28), the acceptance probability for a new sample $z^{(i)}$ for ith iteration is

derived as follows.

$$A(z^{(i)}, z^{(i-1)}) = min\{1, \frac{p(z^{(i)})q(z^{(i-1)}|z^{(i)})}{p(z^{(i-1)})q(z^{(i)}|z^{(i-1)})}\} \tag{2.30}$$

$$= min\{1, \frac{p(z_j^{(i)}|z_{-j}^{(i)})p(z_{-j}^{(i)})p(z_j^{(i-1)}|z_{-j}^{(i)}))}{p(z_j^{(i-1)}|z_{-j}^{(i-1)})p(z_{-j}^{(i-1)})p(z_j^{(i)}|z_{-j}^{(i-1)}))}\} \tag{2.31}$$

$$= 1 \ \ (\because z_{-j}^{(i-1)} = z_{-j}^{(i)}). \tag{2.32}$$

We can see that the acceptance probability is always set to 1 by adopting the proposal distribution in equation (2.29). This means that the samples from gibbs sampler are always accepted and thus, this guarantees the effective convergence rate. However, this method should search whole the dimension of z for every iteration, and this makes it difficult to handle the case when the dimension is large. Also, there are strong dependencies between successive samples because the default Gibbs sampling technique considers one variable at a time. we can improve the basic Gibbs sampler by adopting an intermediate strategy that consecutively extracts samples from groups of variables rather than individual variables. This is achieved in **Blocked Gibbs sampling** [76] selecting blocks of variables and sampling them in turn from the variables in each block and performing the rest of the variables. Futhermore, if we can marginalize out some of the variables, we can priorly integrate out the variables and then conduct the gibbs sampler. This strategy is called **collapsed Gibbs sampling** [77] and [78] is the representative example for the method.

2.4 Optimization Based Methods

2.4.1 Overview

Different from the sampling method, an optimization based method is also an major category in Bayesian approximation. In this method, the varational distribution $q(z)$ approximating the posterior $p(z|x)$ is defined, and the KL-divergence between the two

distribution is minimized.

2.4.2 Kullback-Leibler Divergence

To make the closest approximation $q(z)$ to the posterior distribution $p(z|x)$, we first need to find a way to measure the approximation of the two distributions. The **Kullback-Leibler (KL) divergence** [79] is a widely used measure of the similarity of the two distributions. The KL-divergence between $p(z)$ and $q(z)$ is defined to as

$$\begin{aligned} D_{KL}(q(z)||p(z)) = &\int_z q(z) \log \frac{q(z)}{p(z)} \\ =&E_q[\log \frac{q(z)}{p(z)}]. \end{aligned} \tag{2.33}$$

We note that the KL-divergence is not symmetric in $p(z)$ and $q(z)$. In applications, $p(z)$ typically represents the true distribution of data, observations, or a precisely calculated theoretical distribution, while $q(z)$ typically represents a theory, model, description, or approximation of $p(z)$. The KL-divergence is also called the relative entropy of $p(z)$ with respect to $q(z)$. After introducing the KL-divergence for variational inference, we want to minimize the KL-divergence between the approximate $q(z)$ and the posterior $p(z|x)$. However, in many case, the KL-divergence is intractable and hence, we will solve this problem by introducing the function **evidence lower bound (ELBO)**. As shown in figure 2.5, since the evidence term is constant with respect to $q(z)$, minimizing the KL-divergence term is equivalent to maximizing the ELBO.

2.4.3 Variational Inference

In variational inference, we minimze the KL-divergence between the variational distribution $q(z)$ and the poterior distritubion $p(z|x)$.

$$D_{KL}(q(z)||p(z|x)) = \sum_{z \in Z} q(z) \log \frac{q(z)}{p(z|x)}. \tag{2.34}$$

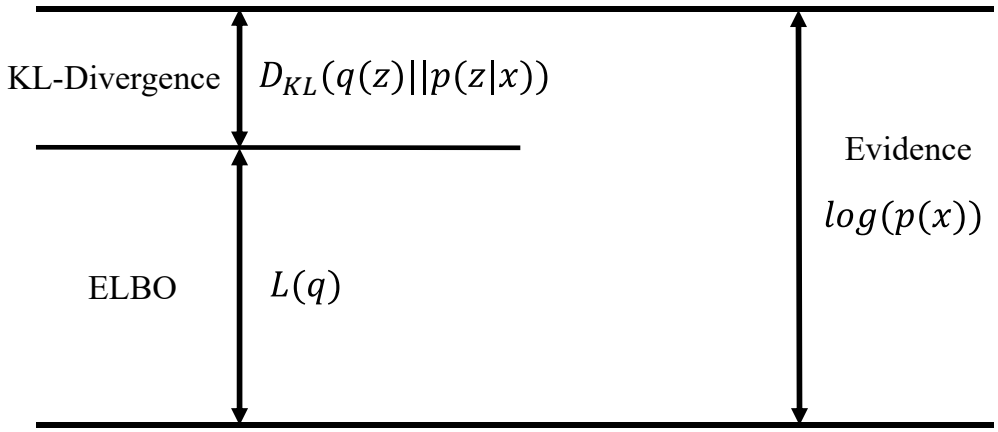


Figure 2.5: A diagram showing the relationship among ELBO, evidence, and KL-divergence. Since the evidence term is constant with respect to $q(.)$, minimizing the KL-divergence term is equivalent to maximizing the ELBO.

The equation can be written as

$$D_{KL}(q(z)||p(z|x)) = \sum_{z \in Z} q(z) \log \frac{q(z)}{p(z,x)} + \log p(x), \tag{2.35}$$

which is same as

$$\begin{aligned} \log p(x) &= D_{KL}(q(z)||p(z|x)) - \sum_{z \in Z} q(z) \log \frac{q(z)}{p(z,x)} \\ &= D_{KL}(q(z)||p(z|x)) + L(q). \end{aligned} \tag{2.36}$$

Since the **evidence** $\log p(x)$ is independent to $q(.)$, the minimizing of $D_K L(q||p)$ is same as maximizing the $L(q)$. The $L(q)$ is called **evidence lower bound (ELBO)**. The problem is that it is also difficult to calculate the ELBO for general $p(z|x)$ and $q(z)$. One way to solve the problem is to apply the **mean field approximation** for $q(z)$.

2.4.4 Mean-Field Approximation

This section describes a family of variational approximation called the **mean field approximation**. In the mean field approximation, we assume that the variational dis-

tribution q for the latent variable z is factorized as

$$q(z_1, z_2, \ldots, z_M) = \prod_{i=1}^{m=M} q_i(z_i) \tag{2.37}$$

Under the mean field approximation above, we can optimize the ELBO using coordinate ascent optimization. Combining equation (2.36) and (2.37), we can get the ELBO as

$$\begin{aligned} L(q) =& E_q[\log p(x,z)] - E_q[\log q(z)] \\ =& E_q[\log p(x,z)] - \sum_{i=1}^{M} E_{q_i}[\log q_i(z_i)] \\ =& E_q[\log p(z|x)] - \sum_{i=1}^{M} E_{q_i}[\log q_i(z_i)] + const. \end{aligned} \tag{2.38}$$

Using chain rule to the posterior distribution $p(x, z)$, we obtain

$$p(z_{1:M}|x) = \prod_{i=1}^{M} p(z_i|z_{1:i-1}, x). \tag{2.39}$$

Combining eqation (2.38) and (2.39), the ELBO is derived as

$$L(q) = \sum_{i=1}^{M} \{E_q[\log p(z_i|z_{1:i-1}, x)] - E_{q_i}[\log q_i(z_i)]\} + const. \tag{2.40}$$

Then, we use the coordinate ascent update for the latent variable z_j with fixing the values of all other latent variables z_{-j}. First, we reorder the latent variables $z_{1:j}$ so that the jth variable comes last., then we take argmax of $L(q)$ with respect to $q(z_j)$. With removing the parts in $L(q)$ that does not depend on $q(z_j)$, we obtain

$$\begin{aligned} argmax_{q_j} L(q) =& argmax_{q_j}(E_q[\log p(z_j|z_{1:j-1}, x)] - E_{q_j}[\log q_j(z_j)]) \\ =& argmax_{q_j}(E_q[\log p(z_j|z_{-j}, x)] - E_{q_j}[\log q_j(z_j)]) \\ =& argmax_{q_j}(\int q_j(z_j) E_{q_{-j}}[\log p(z_j|z_{-j}, x)] dz_j - \int q_j(z_j) \log q_j(z_j) dz_j) \\ =& argmax_{q_j}(L_j). \end{aligned} \tag{2.41}$$

Then we can find argamx $q_j(z_j)$ by taking the derivative of L_j with respect to $q_j(z_j)$. Here we get

$$E_{q_{-j}}[\log p(z_j|z_{-j}, x)] - \log q_j(z_j) - 1 = 0. \tag{2.42}$$

By using equation (2.42), we obtain the coordinate ascent update of the $q_j(z_j)$ as

$$\begin{aligned} q_j^*(z_j) &\propto \exp\{E_{q_{-j}}[\log p(z_j|z_{-j}, x)]\} \\ &\propto \exp\{E_{q_{-j}}[\log p(z_j, z_{-j}, x)]\} \end{aligned} \tag{2.43}$$

In general case of $p(z)$, the expectation in (2.43) is difficult to calculate. However, when we assume the $p(z)$ as exponenetial family [80–83], we can easily evaluate the expectation terms and hence, can analytically update the variational distribution. See [5, 43] for more examples.

2.4.5 Autoencoding Variational Bayes

We have introduced the concept of variational inference so far and have studied the possible solution of the inference using mean field approximation. Using the mean field approximation, we see that the the variational distributions can be updated analytically. However, to use the approximation, the whole model should be designed by exponential family and their conjugate prior, and hence the extent to which data can be described is limited. We can also think of applying Monte Carlo method for approximating $L(q)$, but it was impractical because of the large variance of $q(z)$. However, in recent variants of variational inference algoritm [57] adopt different ways for estimating the variational distribution.

The main process of variatioanl inference is to maximize the ELBO in equation (2.36). From now on we will represent the parameter of joint pdf $p(x, z)$ as θ, and the parameter of variational distribution $q(z)$ as ϕ. In this approach, the variational distribution is defined as conditional distribution $q_\phi(z|x)$ of z given x. Then, the ELBO

is given as

$$L(\theta, \phi; x_i) = -D_{KL}(q_\phi(z|x_i)||p_\theta(z)) + E_{q_\phi(z|x_i)}[\log p_\theta(x_i|z)]. \tag{2.44}$$

Then, our goal is to find θ and ϕ which minimize the ELBO. To solve the problem in general, we can use naive **Monte carlo gradient estimator**, given as

$$\begin{aligned} \bigtriangledown_\phi E_{q_\phi(z|x)}[f(z)] =& E_{q_\phi(z|x)}[f(z) \bigtriangledown_\phi \log q_\phi(z|x)] \\ \propto& \frac{1}{L}\sum_{l=1}^{L} f(z^{(l)}) \bigtriangledown_\phi \log q_\phi(z^{(l)}|x), \end{aligned} \tag{2.45}$$

where $z^{(l)}$ is drawn by $q_\phi(z|x)$. However, as reported in [84], the method is impractical because of very large variance. Therefore, to practically use the strategy, it is crucial to control the amount of the variance.

To control the variance, Kingma [57] uses reparametrization technique to the distribution $q_\phi(z|x_i)$. The reparametrization technique approximates the stochastic distribution $z \sim q_\phi(z|x_i)$ into the differentiable transform $z = g(\epsilon, x_i)$, where $\epsilon \sim p(\epsilon)$. To check the requirements for using the reparametrization, see the material [57].

Using the reparametrization technique, we can control the variance of the sample by selecting proper $p(\epsilon)$. Then the Monte carlo estimate of the expectation is given by

$$\begin{aligned} E_{q_\phi(z|x)}[f(z)] =& E_{p(\epsilon)}[f(g(\epsilon, x))] \\ \propto& \frac{1}{L}\sum_{l=1}^{L} f(g(\epsilon^{(l)}, x)), \end{aligned} \tag{2.46}$$

where $\epsilon^{(l)} \sim p(\epsilon)$. Applying (2.46) to (2.44), we can approximate the equation (2.44) into the eqaution (2.47) using the Monte carlo method with K sample.

$$L(\theta, \phi; x_i) = -D_{KL}(q_\phi(z|x_i)||p_\theta(z)) + \sum_{k=1}^{K} \log p_\theta(x_i|z_{i,k}), \tag{2.47}$$

where $z_{i,k} = g(\epsilon_k, x_i)$ and $\epsilon_k \sim p(\epsilon)$. Experiment [57] reveals that just a few samples K are enough for the approximation. The first term meausres the distance between

the approximate posterior and the prior, while the second term represents an expected negative reconstruction error. This algorithm is valid for any differentiable function $q_\phi(z|x_i)$ satisfying the reparametrization constraint [57] and recent RMSprop [85] or Adam optimizers [70] are frequently applied.

2.5 Gaussian Process Regression

2.5.1 Overview

In section 2.2.3, we have introduced the Bayesian linear regression method which fitting Mth degree polynomial function with M number of coefficient parameters. Now, we describes **Gaussian Process Regression**, which is a nonparametric version of the Bayesian linear regression with Gaussian prior [6].

2.5.2 Weighted Space View

As in equations (2.9-2.12), the prediction model $p(r_*|x_*, \mathbf{x}, \mathbf{r})$ for new data x_*, given input $\mathbf{x} = [x_1, \ldots, x_N]^T$ and $\mathbf{r} = [r_1, \ldots, r_N]^T$, is described by the feature function $\phi(x) = [x\ x^2\ \ldots\ x^M]^T$. We can write (2.9) with matrix form as in by adopting the feature matrix $\Phi = [\phi(x_1) \ldots \phi(x_N)] \in \mathcal{R}^{M \times N}$.

$$p(r_*|x_*, \mathbf{x}, \mathbf{r}) = \mathcal{N}(\rho\phi(x_*)H\Phi\mathbf{r}, \rho^{-1} + \phi(x_*)^T H \phi(x_*)), \tag{2.48}$$

where $H^{-1} = (\alpha I + \rho\Phi\Phi^{-1})$. By applying H into (2.48), we can get

$$\begin{aligned} p(r_*|x_*, \mathbf{x}, \mathbf{r}) = \mathcal{N}(&\phi_*\Sigma_p\Phi(K + \rho^{-1}I)^{-1}\mathbf{r}, \\ &\phi_*\Sigma_p\phi_* - \phi_*^T\Sigma_p\Phi(K + \rho^{-1}I)^{-1}\Phi^T\Sigma_p\phi_*, \end{aligned} \tag{2.49}$$

where $\Sigma_p = \alpha^{-1}I$, $\phi_* = \phi(x_*)$, and $K = \Phi^T\Sigma_p\Phi$. Notice that in (2.49), the feature function is always enters in the form of $\Phi^T\Sigma_p\Phi$, $\phi_*^T\Sigma_p\Phi$ and $\phi_*\Sigma_p\phi_*$. Therefore, we can describe the feature functions by defining the function $k(x, x') = \phi(x)^T\Sigma_p\phi(x')$.

The function is called a **covariance function** or **kernel**. We will see in the next chapter what this function means for GP.

2.5.3 Function Space View

We can see the regression process in different view. In **funtion space view**, we start the regression from assuming the prior of each r_i to be Gaussian process prior, as in

$$\mathbf{r} \sim \mathcal{N}\left(0, \begin{bmatrix} K_{**} & K_*^T \\ K_* & K \end{bmatrix}\right) \tag{2.50}$$

where $\mathbf{r} = [r_*, r_1, ..., r_N]^T$. The matrices K, K_{**} and K_* are defined as

$$K = \begin{bmatrix} k(x_1, x_1) & \cdots & k(x_1, x_N) \\ \vdots & \ddots & \vdots \\ k(x_N, x_1) & \cdots & k(x_N, x_N) \end{bmatrix}, K_{**} = k(x_*, x_*), \tag{2.51}$$

$$K_* = [k(x_*, x_1), k(x_*, x_2), \cdots, k(x_*, k_N)]^T, \tag{2.52}$$

The $k(\cdot, \cdot)$ can be any function that makes the covariance matrix in equation (2.50) to be positive semi-definite. Then, the posterior probability $p(r_*|\mathbf{r}, K, K_*, K_{**})$ is directly solved by schur compliment,

$$r_*|\mathbf{r} \sim \mathcal{N}(K_* K^{-1} \mathbf{r}_{-*}, K_{**} - K_* K^{-1} K_*^T), \tag{2.53}$$

where $\mathbf{r}_{-*} = [r_1, r_2, \cdots, r_N]^T$. We note that the equation (2.53) is equivalent to the equation (2.49). This means that we can design the feature function of Bayesian regression by setting the covariance matrix. For example, assuming the kernel function as an exponential function, it is equivalent to setting the degree of the feature function of Bayesian linear regression to infinity. To see the method for training a hyperparameters of the kernel function, refer to the material [6].

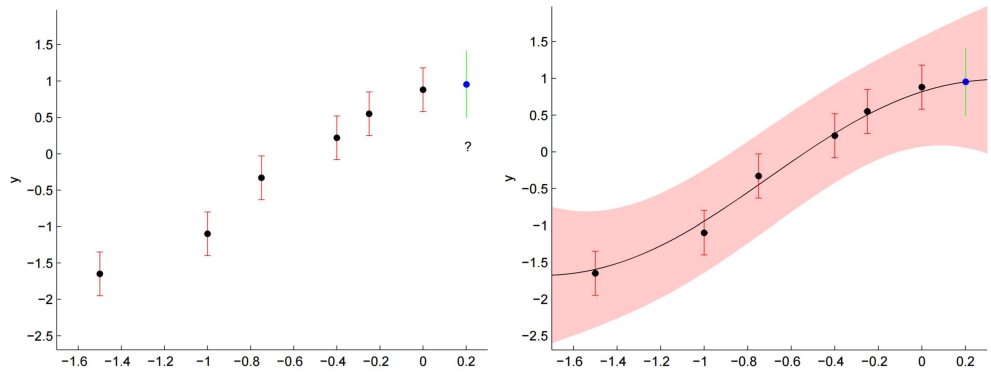


Figure 2.6: Example of Gaussian process regression [86]. In the graph in the left, we have 6 noisy data (red) and want to estimate the response for new input data x_* (blue). The graph in the right describes the estimated responses. Real-line is the mean of the responses and the colored area represents the 95% confidence interval.

Chapter 3

Prediction from Visual Data

3.1 Overall Scheme

The overall scheme of the proposed method is depicted in Figure 3.1. By analyzing the KLT trajectories [87], notable movement patterns are extracted from the scene by the proposed HTGMM. These patterns imply the semantic moving dynamics of objects in a scene, such as going straight or turning right. Therefore, it is natural to expect that some patterns will occur at the same time according to their semantics. For example, we know that straight patterns going right and left in the separated lanes may usually occur simultaneously, as shown in the red lines in Figure 3.1. In this work, we divide the patterns into groups by considering the co-occurrence tendency among them. Each group, therefore, includes the patterns that may occur in the same time span. Utilizing this information, we predict the future trajectory of a target. As seen in Figure 3.1, depending on the dominant group at the prediction time, the predicted path can be different, even if the target starts from the same location. In the below sections, we give a detailed explanation of the proposed method.

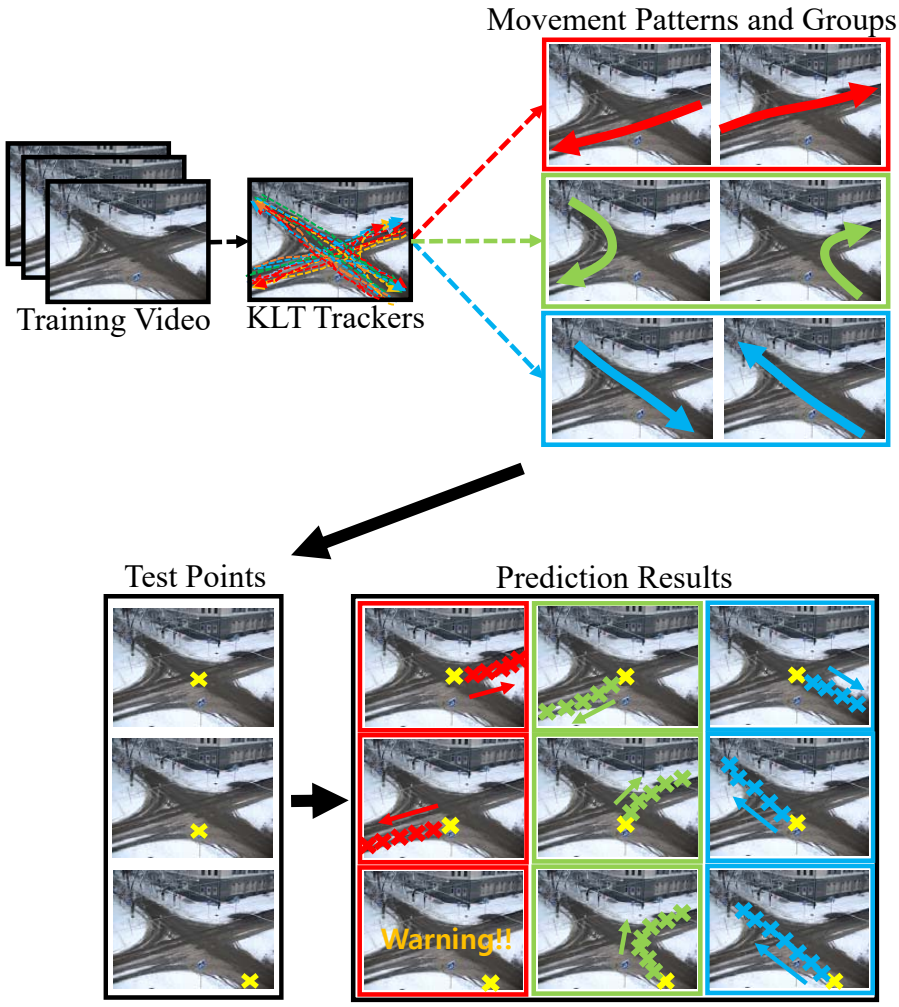


Figure 3.1: Overall framework of the proposed method. The arrow in the scenes refers the movement pattern. Each pattern which occurs at the same time are located in colored boxes. Yellow x point is the location of the target object of which future path be predicted. Depending on the dominant group at prediction time, we induce different predicted paths.

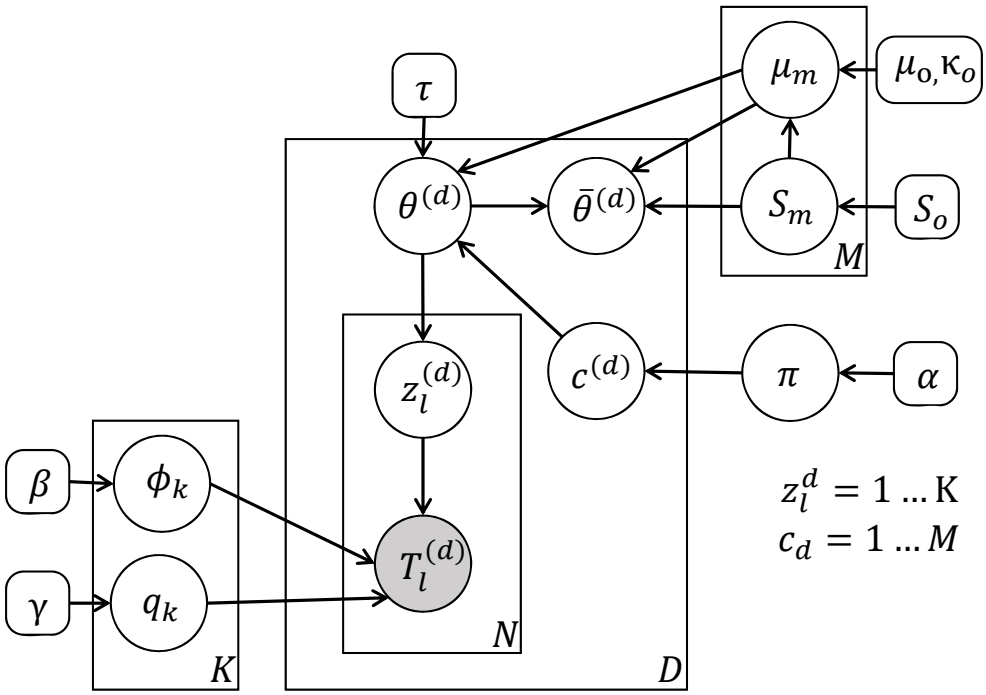


Figure 3.2: The Proposed HTGMM. Each circle represents random variable. Empty circle denotes hidden variable and gray circle is an observed variable. Directed line represents conditional dependency between the circles and rectangle box means that the random variables and their dependency in the box are repeated with the number below the box.

3.2 Conversion of Input Trajectories

First, we convert KLT trajectories [87] into a set of words to be used as input features for the proposed probabilistic model. The sets of KLT trajectories are denoted by $T_l = \{(x_{lt}, y_{lt}) \mid t = 1, ..., N_T\}, l = 1, ..., N$. The term words, $w = \{w_i \mid i = 1, ..., N_w\}$, are defined as indices of the grids dividing a given scene. Then, each point (x_{lt}, y_{lt}) in a trajectory T_l is mapped to the word w_{lt} which indicates the grid including the point. N, N_T, and N_w respectively denote the total number of trajectories, the number of points in each trajectory, and the total number of the words w. Consequently, we can convert the trajectory T_l into the quantized form $T_l^{(w)} = \{w_{lt} \mid t = 1, ..., N_T\}$. In the below sections, we will write the quantized trajectory $T_l^{(w)}$ as T_l for convenience.

3.3 Hierarchical Topic-Gaussian Mixture Model

In this section, we introduce the unsupervised Hierarchical Topic-Gaussian Mixture Model (HTGMM). This model induces typical movement patterns and their co-occurr ence types for a given quantized trajectory T_l. Figure 3.2 illustrates the proposed HTGMM in graphical representation. In a nutshell, the model learns K number of movement patterns into the topic mixture $\{\phi_k, q_k\}, k = 1, ..., K$, by utilizing the quantized KLT trajectories. Then, the patterns are clustered into the mixture of M Gaussians, $\{\mu_m, S_m\}, m = 1, ..., M$, to infer M co-occurrence groups. The following gives the detailed description of the proposed HTGMM.

First of all, to use overall quantized trajectories as input features to the model, we sort all the trajectories in order of ending times of the trajectories and evenly divide them into D number of chunks with N number of trajectories for each chunk. Through this procedure, the trajectories in a chunk occur in similar time span. The whole trajectories $T_l^{(d)}, d = 1, ..., D, l = 1, ..., N$, are used for the observed variables and clustered by the sets of random variables $\{\phi_k, q_k\}, k = 1, ..., K$, which indicates K number of

patterns. $z_l^{(d)}$ is an indexing variable indicating the pattern type of the l-th trajectory in the d-th chunk, ranging from 1 to K. That is, it points out the pattern $\{\phi_k, q_k\}$ including $T_l^{(d)}$ among K patterns. ϕ_k is defined as the N_w dimensional random vector with multinomial distribution. The i-th element of ϕ_k indicates the probability that $k-$th pattern includes the i-th grid location, i.e., i-th word. ϕ_k learns the regional information of the k-th pattern. $q_k \in \mathbb{R}^{N_w \times N_w}$ denotes the word to word transition, i.e., direction, probability of k-th pattern. Consequently, given $z_l^{(d)} = k$, the probability that k-th pattern includes $T_l^{(d)} = \{w_{l1}^{(d)}, w_{l2}^{(d)}, ..., w_{lN_l^{(d)}}^{(d)}\}$, is given by

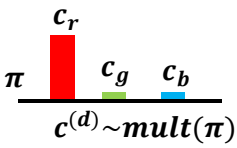
$$p(T_l^{(d)} \mid \phi_k, q_k) = \prod_{j=1}^{N_l^{(d)}} \phi_k(w_{lj}^{(d)}) \prod_{j=1}^{N_l^{(d)}-1} q_k(w_{lj}^{(d)}, w_{l(j+1)}^{(d)}). \tag{3.1}$$

Indexing variable $z_l^{(d)}$ is assigned by the multinomial distribution with parameter $\theta^{(d)} \in \mathbb{R}^K$ as

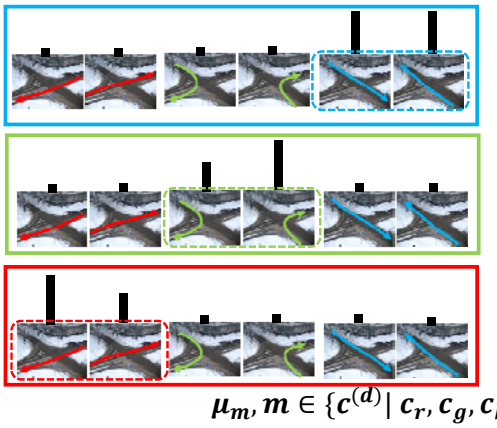
$$z_l^{(d)} \sim mult(\theta^{(d)}), \tag{3.2}$$

where $\sim$ means that the random variable $z_l^{(d)}$ has multinomial distribution with parameter of $\theta^{(d)}$, whereas $\theta^{(d)}$ represents the occurrence frequencies of the patterns in d-th chunk. In $\theta^{(d)}$, the entries with relatively high values give an information that the corresponding patterns have high tendency to occur simultaneously. It means that all $\theta^{(d)}, d = 1, ..., D$, give essential clues to find co-occurrence relationship of patterns. Therefore, we obtain M number of co-occurrence types by grouping $\theta^{(d)}$ into M clusters. To cluster the $\theta^{(d)}$, we set the mixture of M Gaussians $\{\mu_m, S_m\}$, $\mu_m \in \mathbb{R}^K, S_m \in \mathbb{R}^{K \times K}, m = 1, ..., M$. Accordingly, the entries of μ_m with high value represents major patterns in m-th group. The patterns in each group will occur at the same time with high probability. The example of obtained co-occurrence types is shown in Figure 3.3. $c^{(d)}$ is the indexing variable indicating one of Gaussian mixture, ranging from 1 to M. The indexing variable $c^{(d)}$ is assigned by multinomial

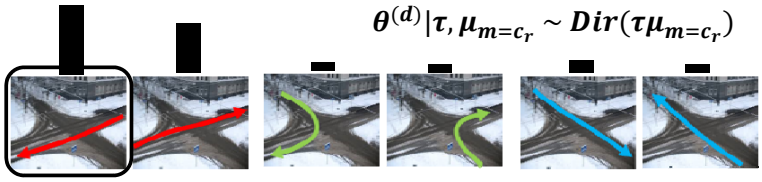
1. Group frequency π



2. Groups of patterns



3. Pattern $\theta^{(d)}$ in time d.



4. Realized pattern and the trajectory

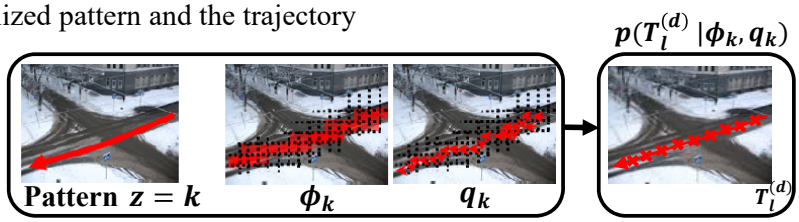


Figure 3.3: Explanation of proposed graphical model. The figure describes the generative procedure of the HTGMM.

distribution with parameter π as

$$c^{(d)} \sim mult(\pi). \tag{3.3}$$

However, since $\{\mu_m, S_m\}$ for the given $c^{(d)} = m$ is not a conjugate prior of $\theta^{(d)}$ [69], the posterior distribution of $\theta^{(d)}$ cannot be easily induced by using $\{\mu_m, S_m\}$ as a Gaussian prior of $\theta^{(d)}$. To resolve the difficulty, we additionally introduce an augmented variable $\bar{\theta}^{(d)} = f(\theta^{(d)})$ where $f(\cdot)$ is a deterministic mapping. It means that $\theta^{(d)}$ is converted to $\bar{\theta}^{(d)}$ with probability one. The performance depending on the choice of the mapping $f(\cdot)$ will be discussed in experiment section. The Gaussian distribution can be the prior of $\bar{\theta}^{(d)}$ with any $f(\cdot)$ because $\bar{\theta}^{(d)}$ is not connected to $z_l^{(d)}$ as shown in Figure 3.2. After that, one of the Gaussian mixture selected by $c^{(d)} = m$ is defined as a prior of $\bar{\theta}^{(d)}$ i.e.,

$$\bar{\theta}^{(d)} \sim \mathcal{N}(\mu_m, S_m). \tag{3.4}$$

Note that $p(\bar{\theta}^{(d)} \mid \mu_m, S_m, \theta^{(d)}) = p(\bar{\theta}^{(d)} \mid \mu_m, S_m)$ given $p(\bar{\theta}^{(d)}|\theta^{(d)}) = 1$.

The procedure in the below is designed to let original $\theta^{(d)}$ assign $z_l^{(d)}$ indicating the dominant pattern in the group, $c^{(d)} = m$. To induce $\theta^{(d)}$ which reflects the co-occurring pattern information μ_m given $c^{(d)} = m$, τ and μ_m is defined as Dirichlet prior of $\theta^{(d)}$ i.e.,

$$\theta^{(d)} \sim Dir(\tau\mu_m). \tag{3.5}$$

Since Dirichlet prior $\tau\mu_m$ is pseudo count [73] of $\theta^{(d)}$, the entry of $\theta^{(d)}$ has higher value as the corresponding entry value of μ_m is larger. Furthermore, it is easy to induce the marginal distribution of $\theta^{(d)}$ because μ_m is conjugate prior of $\theta^{(d)}$. Hyper-parameters $\alpha, \beta, \gamma \in \mathbb{R}$ in Figure 3.2 are Dirichlet prior and $\mu_o \in \mathbb{R}^M, \kappa_o \in \mathbb{R}, S_o \in \mathbb{R}^{M\times M}$ are Nomral-Invert-Wishart prior [73] of Gaussian mixture $\{\mu_m, S_m\}$. These all parameters are conjugate priors of the corresponding random variables.

Joint pdf of the whole model is induced by combining the equations (3.1)-(3.5) altogether. However, it is impossible to get the exact posterior distribution of each variables because integral of the joint pdf is intractable due to the indexing variables z and c. Therefore, approximated inference methods are required to solve the problem. We use Gibbs sampling method [77] for inference of all the hidden variables in the proposed HTGMM.

3.4 Inference of the HTGMM

This section presents the derivation for inferring the hidden variables in the HTGMM in the previos section. To derive the inference procedure in HTGMM, we need to compute the joint pdf of the HTGMM. By considering the dependency among the random variables in the model, the joint pdf can be derived as

$$
\begin{aligned}
&p(\boldsymbol{\phi}, \boldsymbol{q}, \boldsymbol{T}, \boldsymbol{z}, \boldsymbol{\theta}, \bar{\boldsymbol{\theta}}, \boldsymbol{c}, \boldsymbol{\mu}, \boldsymbol{S}, \boldsymbol{\pi} | \alpha, \beta, \gamma, \tau, \mu_o, \kappa_o, S_o) \\
&= \prod_{k=1}^{K} p(\phi_k \mid \beta) p(q_k \mid \gamma) * \\
&[\prod_{d=1}^{D} \{ \prod_{l=1}^{N} p(T_l^{(d)} \mid \phi_{z_l^{(d)}}, q_{z_l^{(d)}}) p(z_l^{(d)} \mid \theta^{(d)}) \} p(\theta^{(d)} \mid \mu_{c^{(d)}}, \tau) p(\bar{\theta}^{(d)} \mid \mu_{c^{(d)}}, S_{c^{(d)}}) p(c^{(d)} \mid \pi)] * \\
&\prod_{m=1}^{M} p(\mu_m \mid S_m, \mu_o, \kappa_o) p(S_m, S_o) * p(\pi \mid \alpha),
\end{aligned}
\tag{3.6}
$$

where the bold character denotes the set of the corresponding elements indexed as in the right-hand side of the equation (3.6). We note that $p(\bar{\theta}^{(d)} \mid \mu_{c^{(d)}}, S_{c^{(d)}}, \theta^{(d)}) = p(\bar{\theta}^{(d)} \mid \mu_{c^{(d)}}, S_{c^{(d)}})$ when $p(\bar{\theta}^{(d)} \mid \theta^{(d)}) = 1$ as mentioned in the paper. To infer the posterior probability for each hidden variable, we should compute an integral to marginalize other variables. However, this equation is not tractable because c, z are natural numbers and the domain of this pdf is not Lebesgue Integrable [88].

Therefore, we use gibbs sampling approach [89] to infer the hidden variables in the proposed HTGMM. The problem is that our model has many random variables

and hence has a large sample space. Accordingly, it is required to reduce the sample space for efficient solving of the problem. To reduce the sample space, we will pre-marginalize out some random variables before the sampling, which is referred to as collapsed gibbs sampling [77]. To utilize the collapsed gibbs sampling method in the proposed HTGMM, we first divide our models into two blocks by using blocked gibbs sampling approach [90]. This method can be applied to our model because the set of variables $\{\boldsymbol{\phi}, \boldsymbol{q}, \boldsymbol{z}\}$ and $\{\boldsymbol{c}, \boldsymbol{\mu}, \boldsymbol{S}, \boldsymbol{\pi}\}$ are conditionally independent given $\boldsymbol{\theta}$. This independency can be easily checked by applying Bayes ball algorithm [91] to the proposed HTGMM. By using the blocked gibbs sampler, we infer the random variables through iteration of the following two steps: *step 1*; update $\{\boldsymbol{\phi}, \boldsymbol{q}, \boldsymbol{z}, \boldsymbol{\theta}\}$ given $\{\boldsymbol{c}, \boldsymbol{\mu}, \boldsymbol{S}, \boldsymbol{\pi}\}$ and *step 2*; update $\{\boldsymbol{c}, \boldsymbol{\mu}, \boldsymbol{S}, \boldsymbol{\pi}\}$ given $\{\boldsymbol{\phi}, \boldsymbol{q}, \boldsymbol{z}, \boldsymbol{\theta}\}$. For each update step, we marginalize all the random variables except the intractable variables $\boldsymbol{z}$ and $\boldsymbol{c}$. We can analytically compute the marginalizing calculation because the random variables are designed to satisfy the conjugate prior by introducing the augmented variable $\bar{\theta}$ as described in the paper. The detailed description of the update procedure is given in the following.

In *step 1*, we will sample only the random variable set $\boldsymbol{z}$. For simplicity, in the below, we will use the redefined notation of $\boldsymbol{T}$ and $\boldsymbol{z}$ by eliminating the chunk index d, that is, $\boldsymbol{z} = \{z_1, ..., z_i, ..., z_{N_o}\}$ and $\boldsymbol{T} = \{T_1, T_2, ..., T_i, ..., T_{N_o}\}$, where N_o indicates the number of all trajectories, i.e., $N_o = N * D$. z_i indicates the assignment variable to assign a pattern index to T_i, and T_i is defined by using the words as $T_i = \{w_{i1}, w_{i2}, ..., , w_{in}, ..., w_{iN_i}\}$, where N_i indicates the number of words in T_i. The chunk including T_i is indexed by d_i. Then, by the Bayes' rule, the conditional posterior distribution for z_i is given by

$$P(z_i = j \mid \boldsymbol{z}_{-i}, \boldsymbol{T}) \propto p(T_i \mid z_i = j, \boldsymbol{z}_{-i}, \boldsymbol{T}_{-i}) P(z_i = j \mid \boldsymbol{z}_{-i}), \tag{3.7}$$

where $\boldsymbol{z}_{-i}$ is the set $\boldsymbol{z}$ excluding z_i, and this notation is also applied to the other variables in the same manner. The first term in the right-hand side in (3.7) is a likelihood,

and the second is a prior. For the first term, we have

$$p(T_i \mid z_i = j, \boldsymbol{z}_{-i}, \boldsymbol{T}) = \int\int p(T_i|z_i = j, \phi_j, q_j) \cdot p(\phi_j, q_j \mid, \boldsymbol{z}_{-i}, \boldsymbol{T}_{-i}) d\phi_j dq_j \tag{3.8}$$

$$= \int\int \prod_{n=1}^{N_i} p(w_{in} \mid z_i = j, \phi_j) * \tag{3.9}$$

$$\prod_{m=1}^{N_i-1} p(w_{i(m+1)} \mid z_i = j, q_j(w_{im}, :)) \cdot p(\phi_j|\boldsymbol{z}_{-i}, \boldsymbol{T}_{-i}) p(q_j|\boldsymbol{z}_{-i}, \boldsymbol{T}_{-i}) d\phi_j dq_j \tag{3.10}$$

$$= \int \prod_{n=1}^{N_i} p(w_{in} \mid z_i = j, \phi_j) p(\phi_j|\boldsymbol{z}_{-i}, \boldsymbol{T}_{-i}) d\phi_j * \tag{3.11}$$

$$\int \prod_{m=1}^{N_i-1} p(w_{i(m+1)} \mid z_i = j, q_j(w_{im}, :)) p(q_j(w_{im}, :)|\boldsymbol{z}_{-i}, \boldsymbol{T}_{-i}) dq_j \tag{3.12}$$

$$= \int \prod_{n=1}^{N_i} p(w_{in} \mid z_i = j, \phi_j) p(\phi_j|\boldsymbol{z}_{-i}, \boldsymbol{w}_{-in}) d\phi_j * \tag{3.13}$$

$$\int \prod_{m=1}^{N_i-1} p(w_{i(m+1)} \mid z_i = j, q_j(w_{im}, :)) p(q_j(w_{im}, :)|\boldsymbol{z}_{-i}, \boldsymbol{w}_{-im}) dq_j \tag{3.14}$$

$$= \int \prod_{n=1}^{N_i} p(w_{in} \mid z_i = j, \phi_j) p(\phi_j|\boldsymbol{z}_{-i}, \boldsymbol{w}_{-in}) d\phi_j * \tag{3.15}$$

$$[\prod_{m=1}^{N_i-1} \int p(w_{i(m+1)} \mid z_i = j, q_j(w_{im}, :)) p(q_j(w_{-im}, :)|\boldsymbol{z}_{-i}, \boldsymbol{w}_{-im}) dq_j(w_{im}, :)]. \tag{3.16}$$

$$(\because \forall q_j(w_s, :) \perp q_j(w_l, :), s \neq l) \tag{3.17}$$

Note that ϕ and q are conditionally independent given T which has been applied to the procedure from (3.8) to (3.9, 3.10). From Bayes' Rule, the second term in (3.15) becomes

$$p(\phi_j|\boldsymbol{z}_{-i}, \boldsymbol{w}_{-in}) \propto p(\boldsymbol{w}_{-in} \mid \phi_j, \boldsymbol{z}_{-i}) p(\phi_j). \tag{3.18}$$

Since $p(\phi_j)$ is *Dirichlet*(β) and conjugate to $p(\boldsymbol{w}_{-in} \mid \phi_j, \boldsymbol{z}_{-i})$, the posterior $p(\phi_j|\boldsymbol{z}_{-i}, \boldsymbol{w}_{-in})$ will be *Dirichlet*($\beta + n^{(w)}_{-in,j}$) as shown in the textbook [72], where $n^{(w)}_{-in,j}$ is the number of instances of word w assigned to pattern j, excluding w_{in}. The first term $p(w_{in} \mid z_i = j, \phi_j)$ in (3.15) is just $\phi^{(j)}_{w_{in}}$ according to the definition of HTGMM. Then, by following the multinomial-Dirichlet prior calculation given in the tutorial [92], we can easily complete the integral in (3.15) with

$$\int \prod_{n=1}^{N_i} p(w_{in} \mid z_i = j, \phi_j) p(\phi_j|\boldsymbol{z}_{-i}, \boldsymbol{w}_{-in}) d\phi_j = \prod_{n=1}^{N_i} \frac{n^{(w)}_{-in,j} + \beta}{n^{(\cdot)}_{-in,j} + W\beta}, \tag{3.19}$$

where W is the total number of words. $n^{(\cdot)}_{-in,j}$ is the total number of instances of all the words in $\boldsymbol{w}$ assigned to pattern j, excluding w_{in}. We can compute the integral in (3.16) using the similar derivation. From Bayes' Rule, the second term in (3.16) becomes

$$p(q_j(w_{in}, :)|\boldsymbol{z}_{-i}, \boldsymbol{w}_{-in}) \propto p(\boldsymbol{z}_{-i}, \boldsymbol{w}_{-in} \mid q_j(w_{in}, :)) p(q_j(w_{in}, :)). \tag{3.20}$$

Subsequently, from the tutorial [92], the posterior $p(q_j(w_{in}, :)|\boldsymbol{z}_{-i}, \boldsymbol{w}_{-in})$ is *Dirichlet*($\gamma + n^{(w)}_{-in}(w_{in})$). The term $n^{(w)}_{-in,j}(w_{in})$ is the number of instances of word w assigned to the transition probability starting from w_{in} for pattern j, excluding the current word w_{in}. By following the same procedure in (3.19), the integral in (3.16) is computed as

$$\begin{aligned} &[\prod_{m=1}^{N_i-1} \int p(w_{i(m+1)} \mid z_i = j, q_j(w_{im}, :)) p(q_j(w_{im}, :)|\boldsymbol{z}_{-i}, \boldsymbol{w}_{-im}) dq_j(w_{im}, :)] \\ &\qquad = \prod_{m=1}^{N_i-1} \frac{n^{(w)}_{-im}(w_{im}) + \gamma}{n^{(\cdot)}_{-im}(w_{im}) + W\gamma}, \end{aligned} \tag{3.21}$$

where $n^{(\cdot)}_{-im}(w_{im})$ is the total number of instances of all the words assigned to the transition probability starting from w_{in} for pattern j, excluding the current word w_{in}. Therefore, from the (3.19),(3.21), the probability $p(T_i \mid z_i = j, \boldsymbol{z}_{-i}, \boldsymbol{T})$ in (3.8) is

derived as

$$p(T_i \mid z_i = j, \boldsymbol{z}_{-i}, \boldsymbol{T}) \propto \prod_{n=1}^{N_i} \frac{n_{-in,j}^{(w)} + \beta}{n_{-in,j}^{(\cdot)} + W * \beta} \prod_{m=1}^{N_i - 1} \frac{n_{-im}^{(w)}(w_{im}) + \gamma}{n_{-in}^{(\cdot)}(w_{im}) + W\gamma}. \tag{3.22}$$

In addition, we can find $p(z_i = j \mid \boldsymbol{z}_{-i})$ in (3.7) with the same procedure as in (3.19). We have

$$\begin{aligned} P(z_i = j \mid \boldsymbol{z}_{-i}) &= \int P(z_i = j \mid \theta^{(d_i)}) p(\theta^{(d_i)} \mid \boldsymbol{z}_{-i}) d\theta^{(d_i)} \\ &= \frac{n_{t_{-i,j}}^{(d_i)} + \tau\mu_c(j)}{n_{t_{-i,\cdot}}^{(d_i)} + K\tau \sum_{k=1}^{K} \mu_c(k)}, \end{aligned} \tag{3.23}$$

when $c^{(d_i)} = c$, because $p(\theta^{(d_i)})$ is defined as $Dirichlet(\tau\mu_c)$. The term $n_{t_{-i,j}}^{(d_i)}$ is the total number of trajectories in chunk d_i assigned to pattern j, excluding the current one. Therefore, from the (3.22),(3.23), the posterior (3.7) is solved as

$$\begin{aligned} &P(z_i = j \mid \boldsymbol{z}_{-i}, \boldsymbol{T}) \\ &\quad \propto \{ \prod_{n=1}^{N_i} \frac{n_{-in,j}^{(w)} + \beta}{n_{-in,j}^{(\cdot)} + W * \beta} \prod_{m=1}^{N_i - 1} \frac{n_{-im}^{(w)}(w_{im}) + \gamma}{n_{-in}^{(\cdot)}(w_{im}) + W\gamma} \} \{ \frac{n_{t_{-i,j}}^{(d_i)} + \tau\mu_c(j)}{n_{t_{-i,\cdot}}^{(d_i)} + K\tau \sum_{k=1}^{K} \mu_c(k)} \}. \end{aligned} \tag{3.24}$$

We highlight that this derivation is possible by employing the augment variable $\bar{\theta}$ of which prior is the Gaussian distribution $\mathcal{N}(\mu_c, S_c)$. If we naively define the prior of $\theta^{(d)}$ as $\mathcal{N}(\mu_c, S_c)$, the integral in (3.23) is intractable because the Gaussian distribution is not a conjugate prior for the multinomial $\theta^{(d)}$. However, since we employ $\bar{\theta}^{(d)}$ which is given by deterministic mapping from $\theta^{(d)}$ and make $\bar{\theta}^{(d)}$ have the Gaussian prior, we can let $\theta^{(d)}$ has Dirichlet prior satisfying the conjugate prior. In *step 2*, we compute update equation considering both $\theta^{(d)}$ and $\bar{\theta}^{(d)}$.

For *step 2*, we will sample only $c^{(d)}$, the assignment of the $\theta^{(d)}$, to infer the hidden variables $\{\boldsymbol{\mu}, \boldsymbol{S}, \boldsymbol{\pi}\}$. Similar to the equation (3.7), we compute the posterior distribu-

tion for $c^{(d)}$ as

$$
\begin{aligned}
&P(c^{(d)} = c \mid \boldsymbol{c}_{-d}, \bar{\boldsymbol{\theta}}, \boldsymbol{\theta}) \\
&\propto P(c^{(d)} = c \mid \boldsymbol{c}_{-d}) p(\bar{\boldsymbol{\theta}}, \boldsymbol{\theta} \mid c^{(d)} = c, \boldsymbol{c}_{-d}) \\
&= P(c^{(d)} = c \mid \boldsymbol{c}_{-d}) p(\bar{\theta}^{(d)}, \theta^{(d)} \mid \bar{\boldsymbol{\theta}}_{-d}, \boldsymbol{\theta}_{-d}, c^{(d)} = c, \boldsymbol{c}_{-d}) p(\bar{\boldsymbol{\theta}}_{-d}, \boldsymbol{\theta}_{-d} \mid c^{(d)} = c, \boldsymbol{c}_{-d}) \\
&\propto P(c^{(d)} = c \mid \boldsymbol{c}_{-d}) p(\bar{\theta}^{(d)}, \theta^{(d)} \mid \bar{\boldsymbol{\theta}}_{-d}, \boldsymbol{\theta}_{-d}, c^{(d)} = c, \boldsymbol{c}_{-d}).
\end{aligned}
\tag{3.25}
$$

The equation (3.25) is further derived as

$$
\begin{aligned}
&P(c^{(d)} = c \mid \boldsymbol{c}_{-d}, \bar{\boldsymbol{\theta}}, \boldsymbol{\theta}) && (3.26)\\
&\propto P(c^{(d)} = c \mid \boldsymbol{c}_{-d}) p(\bar{\theta}^{(d)}, \theta^{(d)} \mid \bar{\boldsymbol{\theta}}_{-d}, \boldsymbol{\theta}_{-d}, c^{(d)} = c, \boldsymbol{c}_{-d}) && (3.27)\\
&\propto P(c^{(d)} = c \mid \boldsymbol{c}_{-d}) p(\bar{\theta}^{(d)} \mid \bar{\boldsymbol{\theta}}_{-d}, \boldsymbol{\theta}_{-d}, c^{(d)} = c, \boldsymbol{c}_{-d}, \theta^{(d)}) p(\theta^{(d)} \mid \bar{\boldsymbol{\theta}}_{-d}, \boldsymbol{\theta}_{-d}, c^{(d)} = c, \boldsymbol{c}_{-d}) && (3.28)\\
&\propto P(c^{(d)} = c \mid \boldsymbol{c}_{-d}) p(\bar{\theta}^{(d)} \mid \bar{\boldsymbol{\theta}}_{-d}, c^{(d)} = c, \boldsymbol{c}_{-d}) p(\theta^{(d)} \mid \bar{\boldsymbol{\theta}}_{-d}, c^{(d)} = c, \boldsymbol{c}_{-d}), && (3.29)\\
&(\because p(\bar{\theta}^{(d)} | \theta^{(d)}) = 1). && (3.30)
\end{aligned}
$$

By using the same derivation step with (3.19), the first term in (3.29) is given by

$$
P(c^{(d)} = c \mid \boldsymbol{c}_{-d}) = \frac{n_{m-d,c} + \alpha}{n_{m-d,(\cdot)} + M\alpha}, \tag{3.31}
$$

Since $\bar{\theta}^{(d)}$ is drawn from Gaussian distribution, the second term in (3.29) is equivalent to Gaussian posterior distribution. Accordingly, by following the tutorial [72, 93], the second term is given as

$$
\begin{aligned}
p(\bar{\theta}^{(d)} \mid &\bar{\boldsymbol{\theta}}_{-d}, c^{(d)} = c, \boldsymbol{c}_{-d}) = \\
&\zeta(\bar{\theta}^{(d)} \mid \mu_{-d,c}, \frac{\kappa_n + 1}{\kappa_n(\upsilon_n - D + 1)} S_{-d,c}, \upsilon_n - D + 1),
\end{aligned}
\tag{3.32}
$$

where $\zeta(\cdot)$ is standard-t distribution. The $\mu_{-d,c}, S_{-d,c}, \kappa_n$ and υ_n are given by

$$\begin{aligned}
\mu_{-d,c} &= \frac{\kappa_o \mu_o + \sum_{d=1}^{D} \bar{\theta}^{(d)} I(c^{(d)} = c)}{\kappa_n}, \\
\kappa_n &= \kappa_o + \sum_{d=1}^{D} I(c^{(d)} = c), \\
\upsilon_n &= \upsilon_o + \sum_{d=1}^{D} I(c^{(d)} = c), \\
S_{-d,c} &= S_o + S_c + \kappa_o \mu_o \mu_o^T - \kappa_n \mu_{-d,c} \mu_{-d,c}^T, \\
S_c &= \sum_{d=1}^{D} \bar{\theta}^{(d)} \bar{\theta}^{(d)^T} I(c^{(d)} = c),
\end{aligned} \tag{3.33}$$

where $I(\cdot)$ is an indicator function. As defined in our paper, the third term in (3.29) is Dirichlet distribution over $\tau\mu_{-d,c}$ and so given as

$$p(\theta^{(d)} \mid \bar{\boldsymbol{\theta}}_{-d}, c^{(d)} = c, \boldsymbol{c}_{-d}) = Dirichlet(\theta^{(d)} \mid \tau\mu_{-d,c}). \tag{3.34}$$

Therefore, from (3.31),(3.32),(3.34), the posterior equation (3.26) is solved as

$$\begin{aligned}
&P(c^{(d)} = c \mid \boldsymbol{c}_{-d}, \bar{\boldsymbol{\theta}}, \boldsymbol{\theta}) \\
&\quad \propto \frac{n_{m-d,c} + \alpha}{n_{m-d,(\cdot)} + M\alpha} \cdot \zeta(\bar{\theta}^{(d)} \mid \mu_{-d,c}, \frac{\kappa_n + 1}{\kappa_n(\upsilon_n - D + 1)} S_{-d,c}, \upsilon_n - D + 1) * \\
&\quad Dirichlet(\theta^{(d)} \mid \tau\mu_{-d,c}).
\end{aligned} \tag{3.35}$$

By iteratively resampling $\boldsymbol{z}$ and $\boldsymbol{c}$ by the equations (3.24) and (3.35), we can infer the hidden variables of the proposed HTGMM.

3.5 Deterministic Method for Path Prediction

This section presents the path prediction method using the movement patterns and their co-occurrence groups learned by the proposed HTGMM. For this, we have to resolve two main problems. The first problem is that the movement patterns are described in quantized space. The other problem is that transition probability among words are

defined only in the area of learned patterns. Therefore, we first suggest a method to expand the transition information of q_k into the entire word pairs. Then, we propose the final path prediction method inducing the future location x_{t+1} at time t in continuous domain given a previous target path $\mathbf{x}_t = \{x_1, x_2, ..., x_t\}$ in an iterative manner.

Relaxation of word to word transition: The word to word transition $q_k(w_i, w_j)$ indicates the direction of k-th movement pattern from i-th grid to j-th grid. The (w_i, w_j) is a word pair in a scene where the condition $q_k(w_i, w_j) \neq 0$ is satisfied. Since we do not have the transition information for all the word pairs, the total number of trained word pairs (w_i, w_j) is less then whole possible number of word pairs ${N_w}^2$. To expand the word to word transitions to whole word pairs, we employ an energy potential vector $\mathbf{y} = [y_1, y_2, ..., y_{N_w}]^T$. The y_i, y_j are defined so that $y_i - y_j = q_k(w_i, w_j)$. If we know the transition probabilities for R pairs of words, we can set R equations for each (y_i, y_j). The set of the equations can be expressed by sparse matrix form $A\mathbf{y} = b, A \in \mathbb{R}^{R \times N_w}, b \in \mathbb{R}^R$ which holds $A[r, i] = 1$,$A[r, j] = -1$ and $b[r] = q_k(w_i, w_j)$. $A[r, i]$ and $A[r, j]$ are (r, i) and (r, j) element of matrix A. Also, $b[r]$ is the r-th element of vector b. In most cases, A is not a full rank matrix. Accordingly, we can find a solution as $\mathbf{y} = (A^T A)^{-1} A^T \mathbf{b}$ by using pseudo inverse. Using the $\mathbf{y}$, we induce transition probabilities of whole word pairs in a scene. Figure 3.4 is an example illustrating the $\mathbf{y}$. The difference between y_i and y_j at each location denotes the possibility that the target moves from high potential position w_i to low potential position w_j. The Figure 3.4 shows that the potential value decreases as the target moves to the future locations.

Path Prediction in Continuous Domain: After finding the potential map $\mathbf{y}$, we iteratively update $\mathbf{x}_t$. The overall path prediction procedure has three steps. In the first step, we find the movement patterns adequate to the target object by using the inference results from HTGMM. The second step is the updating procedure for $\mathbf{y}_t$. In this step, we modify $\mathbf{y}$ to reflect the past trajectory of the target, $\mathbf{x}_t$. We denote $\mathbf{y}$ at time t as $\mathbf{y}_t$.

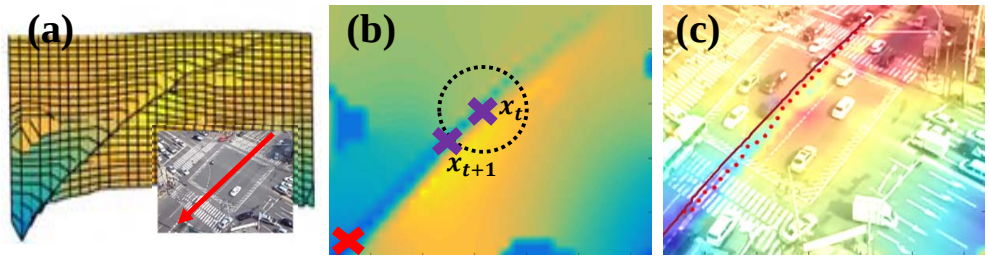


Figure 3.4: The result of expanded word to word transition. We obtain energy potential map in (a) by expanding word to word transition of the pattern in bottom of (a). The potential goes down from yellow to blue. We induce the potential map in continuous domain by bi-lnear interpolation in (b). Therefore, the sink point of the map (red 'x') indicates the destination of the pattern. The points at purple 'x' represent x_{t+1} and x_t. The figure (c) shows the example path prediction result.

In the last step, we estimate future location x_{t+1} of the target using the updated $\mathbf{y}_{t+1}$. *(1) Pattern selection step:* This step begins with converting the past trajectory $\mathbf{x}_t = \{x_i \mid i = 1, ..., t\}$ into a quantized form $\mathbf{x}_t^{(q)} = \{w_i \mid i = 1, ..., t\}$ where w_i is a word including x_i. Then we select the pattern including $x_t^{(q)}$ according to the probability of selecting k-th pattern $\{\phi_k, q_k\}$ given the dominant pattern group c by employing the results from HTGMM as

$$p(\{\phi_k, q_k\} \mid \mathbf{x}_t^{(q)}, \mu_c) \propto p(\mathbf{x}_t^{(q)} \mid \{\phi_k, q_k\}) p(z = k \mid \mu_c). \tag{3.36}$$

The first term in the right-hand side of the equation can be obtained from the equation (3.1). It represents the probability that k-th movement pattern includes the target trajectory $\mathbf{x}_t$. The second term is a Dirichlet multinomial distribution over μ_c. The distribution is induced by marginalizing θ of $p(z = k \mid \theta)p(\theta \mid \mu_c)$ where $p(z = k \mid \theta)$ and $p(\theta \mid \mu_c)$ can be obtained from the equations (3.2) and (3.5). It is a tractable calculation because μ_c is a conjugate prior for θ. The second term leads to the selection of

z indicating the frequently occurring pattern in the group c. The group c is determined by the maximum value of the posterior probability for μ_c in the HTGMM with given co-occurring KLT trajectories.

(2). Energy potential map update step: After selecting the pattern k, we update $\mathbf{y}_{t+1}^k$ using $\mathbf{y}_t^k$ and $\mathbf{x}_t$. We denote $\mathbf{y}_t^k$ as the potential vector $\mathbf{y}$ for k-th pattern at time t. To estimate $\mathbf{y}_{t+1}^k$ reflecting the trace $\mathbf{x}_t$, we first define $t-1$ terms in equation (3.37) from the $\mathbf{x}_t^{(q)}$, where y_i is the energy potential assigned for the word w_i in $\mathbf{x}_t$.

$$y_{w_{i+1}} - y_{w_i} = p, i = 1, ..., t-1. \tag{3.37}$$

Then, we add them into the rows of the matrix A, b used previously for calculating the potential vector. By solving the linear equation $A\mathbf{y} = b$ with modified A and b, we obtain a new vector $\mathbf{y}_c$ containing the future dynamics estimated from the past movements. We set p as a mean value of all $q_k(w_u, w_v) \geq 0$ in a scene. The vector $\mathbf{y}_{t+1}^k$ is updated by reflecting the $\mathbf{y}_c$ to the present state as

$$\mathbf{y}_{t+1}^k = (1-\alpha)\mathbf{y}_t^k + \alpha\mathbf{y}_c, \tag{3.38}$$

where term α is a design parameter.

(3) Path prediction step: Now we finally find $\mathbf{x}_{t+1}$ using the $\mathbf{y}_{t+1}^k$. As seen in Figure 3.4, the map $\mathbf{y}_{t+1}^k$ forms a valley-like shape going down to the destination of the pattern k. Therefore, we find x_{t+1} by following the slope of the valley. To find new x_{t+1} in a continuous domain, we expand $\mathbf{y}_{t+1}^k$ into continuous space using bi-linear mapping [94]. $\mathbf{F}_{t+1}^k$ refers the continuous energy potential map obtained from $\mathbf{y}_{t+1}^k$. Then, we find the sink point x_s of the $\mathbf{F}_{t+1}^k$ which indicates the destination of the pattern k. The optimization formulation to find x_{t+1} is given by

$$\begin{aligned} x_{t+1} =& min_x \mathbf{F}_{t+1}^k(x), \\ s.t.\ & \|x - x_t\|_2 = \|x_t - x_{t-1}\|_2, \\ & \|x - x_s\|_2 \leq \|x_t - x_s\|_2. \end{aligned} \tag{3.39}$$

To find the minimal point in (3.39), we only need to navigate the points x lying in the circle $C(R, \theta)$ which $R = \|x_t - x_{t-1}\|_2$ and $-\pi \leq \theta \leq \pi$ with center x_t. To find x with minimal $F_{t+1}^k(x)$, we find inflection points by calculating θ satisfying the gradient $\nabla_\theta F_{t+1}^k(C(R, \theta)) = 0$ and choose the point with the minimum field value as the future location x_{t+1}. By increasing the time index t, we predict the future location of target recursively and we terminate the recursive iteration when the distance between predicted point x_{t+1} and x_s is smaller than $\|x_t - x_{t-1}\|_2$ or x_{t+1} goes over the boundary of the scene.

Chapter 4

Regression of Visual Data

4.1 Overall Scheme

Given the target data pair $(x_i, y_i), i = 1, \ldots, N, x_i \in \mathcal{X}, y_i \in \mathcal{Y}$, our goal is to find the unknown response y_* for a new input x_*. In this paper, the response $y \in \mathcal{Y}$ is defined as an image and the corresponding input $x \in \mathcal{X}$ is defined accordingly based on the applications as in Fig. 1.1.

As shown in Fig. 4.1, for the observed data pairs $(x_i, y_i), i = 1, \cdots, N$, the encoder/decoder produces $\hat{y}_i$ which is the reconstruction of an observed image y_i. For the observed data, the encoding network $E(\cdot)$ produces mean and variance for a part of the latent vector z_i, that is, $[m_{i,y}, \sigma_{i,y}] = E(y_i; W_E)$ which compresses y_i to a latent variable with Gaussian mean $m_{i,y}$ and variance $\sigma_{i,y}$. The remaining part of z_i is modeled by $[m_{i,x}, \sigma_{i,x}] = f(x_i, W_x)$ which represents mean and variance of x_i. Thus, the Gaussian distribution of z_i is described by $m_i = [m_{i,y}, m_{i,x}]$ and $\sigma_i = diag[\sigma_{i,y}, \sigma_{i,x}]$. For the unobserved image y_* for a newly given x_*, the proposed method produces $\hat{y}_*$, which is an estimator of y_*.

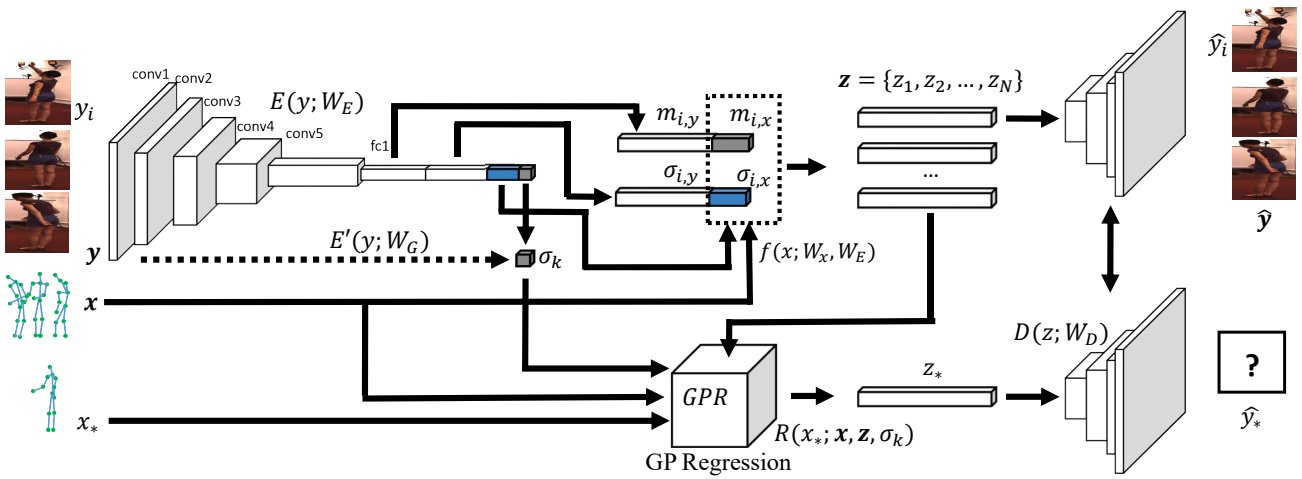
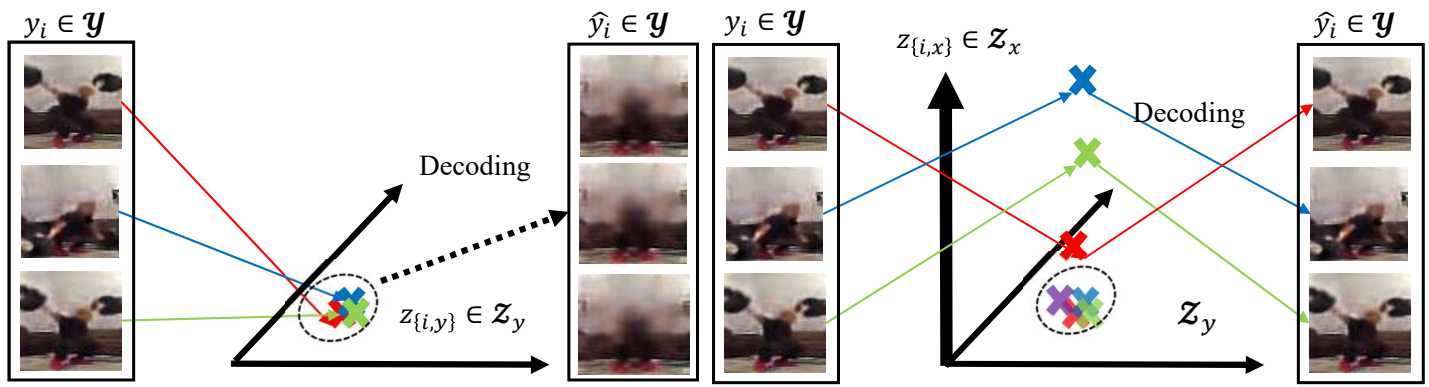


Figure 4.1: Overall scheme of the proposed method. For observed data pairs $\mathbf{x} = \{x_1, x_2, \cdots, x_N\}$ and $\mathbf{y} = \{y_1, y_2, \cdots, y_N\}$, the proposed autoencoder reconstructs $\hat{y_i} \simeq y_i$ as shown in the top right. For the unobserved y_* to given x_*, it is impossible to obtain z_* by using an encoder with W_E, because we do not have information about y_*. Thus to estimate y_*, we obtain z_* using regression from $\mathbf{x}$, $\mathbf{z}$ and x_*, and estimate the response $\hat{y}_*$ from z_*.

The idea of the additional domain knowledge on x_i for defining $z_i \in \mathcal{Z}$ is especially beneficial when capturing delicate changes of an object in the case of a fixed background. For example, without adding domain knowledge $[m_{i,x}, \sigma_{i,x}]$, the diverse postures occurring in the same background are mapped to similar locations in the latent space as shown in Fig. 4.2 (a) , which results in the unsatisfactory reconstructions of in-distinctive action images. Instead, by adding the domain axis to the $\mathcal{Z}$ as shown in Fig. 4.2 (b), we can separate each projection z_i from others.

Using z_i sampled from $\mathcal{N}(m_i, \sigma_i)$, the decoding network reconstructs the output response $\hat{y}_i$, that is, $\hat{y}_i = D(z_i; W_D)$. Note that if (W_E, W_x, W_D) is well trained by the training scheme in Section 4.4, $\hat{y}_i$ should be similar to y_i. However, for an unobserved y_* to a given x_*, it is impossible to obtain z_* from $E(\cdot; W_E)$ because we do not have any information on y_*. To estimate y_*, we obtain z_* by using regression from $\mathbf{x} = \{x_1, x_2, \cdots, x_N\}, \mathbf{z} = \{z_1, z_2, \cdots, z_N\}$ and x_*. For this regression, z_i is sampled from $\mathcal{N}(m_i, \sigma_i)$ for each observed response $y_i \in \mathbf{y} = \{y_1, y_2, \cdots, y_N\}$. Then, we estimate z_* using Gaussian process (GP) regression $z_* \sim R(x_*; \mathbf{x}, \mathbf{z}, \sigma_k)$ to be described in Section 4.2, where σ_k is a kernel parameter of the GP regression, which can be produced by an additional encoder $\sigma_k = E'(\mathbf{y}, W_G)$; with $\sigma_k = [\sigma_{k,1}, \cdots, \sigma_{k,N}]$. In this paper, for computational simplicity, we combine this kernel encoder with the encoder network $E(y; W_E)$ and change the outputs into $[m_{i,y}, \sigma_{i,y}, \sigma_{i,k}] = E(y_i, W_E)$.



(a) With domain space

(b) Without domain space

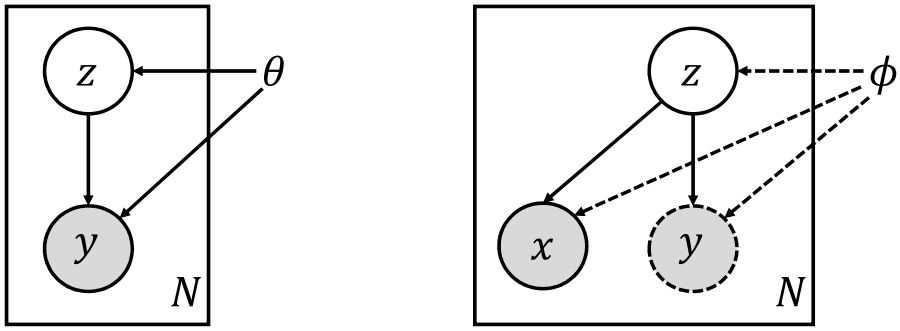
Figure 4.2: Configuration of the latent space $\mathcal{Z}$. By adding the additional space regarding the domain $\mathcal{X}$, the latent space can separate the responses y_i occurred by different x_i.

After z_* is estimated, the response $\hat{y}_*$ is reconstructed from z_* by using the decoding network $D(z_*; W_D)$. Note that the $D(z_*; W_D)$ should reconstruct not only $\hat{y}$ from the z_i sampled by $\mathcal{N}(m_i, \sigma_i)$, but also y_* from the z_* which is the regression result obtained from $x_*, \mathbf{x}$, and $\mathbf{y}$. The whole procedure is designed as a generative framework with joint distribution $p(x_*, y_*, \mathbf{x}, \mathbf{y}, W_E, W_x, W_D)$, and hence can be derived by the VAE algorithm.

4.2 Variational Autoencoded Regression

The proposed scheme (depicted in Fig. 4.1) is derived from the directed graph model in Fig. 4.3. The diagram in Fig. 4.3 (a) represents the generative model describing a typical reconstruction problem, and the diagram in Fig. 4.3 (b) is the variational model which not only approximates the generative model in Fig. 4.3 (a), but also performs the regression for the estimation of unobserved y by utilizing an information variable x related to y. The joint distribution $p_\theta(y, z)$ can be expressed by the likelihood function $p_\theta(y|z)$ and the prior distribution $p_\theta(z)$, where θ refers to the set of all parameters related to the generation of the response $y \in \mathcal{Y}$ from the latent variable z. In our method, the prior distribution of z is defined as zero mean Gaussian distribution, as in typical variants of VAE [57, 63]. Also, the likelihood function $p_\theta(y|z)$ depicts the decoding process in the proposed scheme. Below, it is shown that θ is realized by the parameter W_D of the decoding network.

Once the joint distribution $p_\theta(y, z)$ is defined, the posterior $p_\theta(z|y)$ can be theoretically derived from the Bayes theorem, but the calculation is intractable. Therefore, the variational distribution $q_\phi(z|x, y)$ is introduced to approximate the true posterior distribution $p_\theta(z|y)$. Unlike $p_\theta(z|y)$, x is introduced for the variational distribution $q_\phi(z|x, y)$ to sample $z_* \sim R(x_*; \mathbf{x}, \mathbf{z}, \sigma_k)$, which is the result of the GP regression for the unknown y_*. $q_\phi(z|x, y)$ represents the overall encoding procedure generating the



(a) Generative model $p_\theta(y|z)p_\theta(z)$ (b) Variational model $q_\phi(z|x,y)$

Figure 4.3: The directed graphical model of the proposed method. (a) Generative model for y and the latent variable z. (b) Variational distribution to approximate the posterior $p_\theta(z|y)$ of the generative model. y is not observed for the newly given input $x = x_*$.

latent variable z from the input data pair (x, y) and correspondingly, the variational parameter ϕ is realized by the parameters W_E, W_x as described in Section 4.1. Importantly, $q_\phi(z|x, y)$ should be able to explain both cases: 1) an observed image y_i, and 2) an unobserved image y_* which requires regression as mentioned previously. For the first case, the variational distribution is defined as $z_i \sim q_\phi(z|x = x_i, y = y_i)$. For the latter case, the variational distribution is defined as $z_* \sim q_\phi(z|x = x_*, y \in \varnothing)$ which represents the GP regression procedure for estimating latent z_* for the input x_*.

In order to estimate the parameters θ and ϕ which minimize the distance between $p_\theta(z|y)$ and $q_\phi(z|x,y)$, we minimize the Kullback−Leibler divergence $D_{KL}(p_\theta(z|y)||$ $q_\phi(z|x,y))$. Following the derivation in [57,95], the minimization procedure $\{\theta^*,\phi^*\} = \arg\min_{\{\theta,\phi\}} D_{KL}(p_\theta(z|y)||q_\phi(z|x,y))$ is converted to $\{\theta^*,\phi^*\} = \arg\min_{\{\theta,\phi\}} L(\theta,\phi)$, where

$$\begin{aligned} L(\theta,\phi) = &- D_{KL}(q_\phi(z|x,y)||p_\theta(z)) \\ &+ \sum_{i=1}^{N} \log p_\theta(y_i|z_i) + \sum_{j=1}^{M} \log p_\theta(\hat{y}_{*j}|z_{*j}). \end{aligned} \tag{4.1}$$

$z_{*,j}$ and $y_{*,j}$ represents the M number of latent codes and output responses for $x_{*,j}, j = 1,\cdots,M$, to be regressed. In (4.1), $\hat{y}_{*,j} = D(z_{*,j};W_D)$ is the reconstructed response from $z_{*,j}$ by the decoding network as depicted in Fig. 4.1. The parameters θ and ϕ are realized by the connection parameters of the encoding network with regression for $q_\phi(z|x,y)$, and the decoding network for $p_\theta(z|y)$ (see Section 4.3). To minimize the loss in (4.1), we propose a method for mini-batch learning (see Section 4.4). The Adam optimizer [70] is used for stochastic gradient descent training.

4.3 Model Description

For the encoding part, we define $q_\phi(z|x,y)$ which maps the data pair (x,y) into the latent space $\mathcal{Z}$. For both, observed and unobserved images, $q_\phi(z|x,y)$ is defined by Gaussian distribution as in (4.2) and it enables us to analytically solve the KL-divergence term $D_{KL}(q_\phi(z|x,y)||p_\theta(z))$ in (4.1) following [57]:

$$q_\phi(z|x,y) = \mathcal{N}(z|m(x,y),\sigma(x,y)). \tag{4.2}$$

The variational parameter ϕ consists of the Gaussian mean function $m(x,y)$ and the variance function $\sigma(x,y)$. The $m(x,y)$ and $\sigma(x,y)$ are produced in different ways depending on the input data. When the input data is given by $x = x_i \in \mathbf{x}$, the encoder

yields $m(x,y) = [m_{i,y}, m_{i,x}]$ and $\sigma(x,y) = diag[\sigma_{i,y}, \sigma_{i,x}]$, where $diag[\cdot]$ refers to a diagonal matrix. When the input data is given by $x = x_{*,j} \in \mathbf{x}_*$, $m(x,y)$ and $\sigma(x,y)$ are determined by the mean and variance $(m_{*,j}, \sigma_{*,j})$ estimated by GP regression from **z**, **x** and $x_{*,j}$, where

$$m_{*,j} = K_{*,j}K^{-1}\mathbf{Z},\ \sigma_G = (K_{**,j} - K_{*,j}K^{-1}K_{*,j}^T)I. \tag{4.3}$$

Z refers to the matrix $[z_1; z_2; \cdots; z_N] \in \mathcal{R}^{N\times D}$, and $I \in \mathcal{R}^{D\times D}$ is the identity matrix, where D is the dimension of $z \in \mathcal{Z}$. The matrices $K, K_{**,j}$ and $K_{*,j}$ are defined as

$$K = \begin{bmatrix} k(x_1,x_1) & \cdots & k(x_1,x_N) \\ \vdots & \ddots & \vdots \\ k(x_N,x_1) & \cdots & k(x_N,x_N) \end{bmatrix}, \tag{4.4}$$

$$K_{**,j} = k(x_{*,j}, x_{*,j}), \tag{4.5}$$

$$K_{*,j} = [k(x_{*,j},x_1), k(x_{*,j},x_2), \cdots, k(x_{*,j},x_N)]. \tag{4.6}$$

For the kernel $k(\cdot,\cdot)$, we use a simplified version of SE-kernel [6], where $k(x_i,x_j) = \sqrt{\sigma_i\sigma_j}\exp||x_i - x_j||^2$. Eventually, the variational parameter ϕ is realized by the weight matrices (W_x, W_E) of the encoder network. In summary, for the given data **x**, **y**, and $\mathbf{x}_*$, $q_\phi(z|x,y)$ in (4.2) is given as

$$q_\phi(z|x,y) = \begin{cases} \mathcal{N}(m_i,\sigma_i) & x = x_i, y = y_i. \\ \mathcal{N}(m_{*,j},\sigma_{*,j}) & x = x_{*,j}, y \in \varnothing. \end{cases} \tag{4.7}$$

For the decoding procedure, we define the likelihood function $p_\theta(y|z) = p(y|D(z;W_D))$, where $p(y|D(z;W_D))$ is defined as a Gaussian distribution with mean $D(z;W_D)$ and fixed variance. Since the prior of z is defined with zero mean Gaussian and identity covariance matrix, the weight W_D represents the generative model parameter θ. Correspondingly, the meanings of the second term and third term in (4.1) are interpreted as follows. Since the negative log-likelihood $(-\log(p_\theta(y|z)))$ is defined

as l_2 distance $||y - D(z; W_D)||^2$ in our algorithm, the second term represents the reconstruction error for the given data pair (x_i, y_i) to y_i, and the third term denotes the estimation error for $y_{*,j}$ via regression from the given input data $x_{*,j}$ and the observed data $(x_i, y_i), i = 1, \ldots, N$.

4.4 Training

To train the parameters of the proposed model, a sufficient number of the training datasets is required. In our algorithm, a total of V different training sequences (x_i^v, y_i^v), $v = 1, \ldots, V, i = 1 \ldots, N_v$ are used, as shown in Fig. 4.4. These training data pairs share similar semantics to the target (test) data pair (x_i, y_i). If the target data pair is a golf swing sequence, the training data pairs will be different golf swing sequences obtained in different situations.

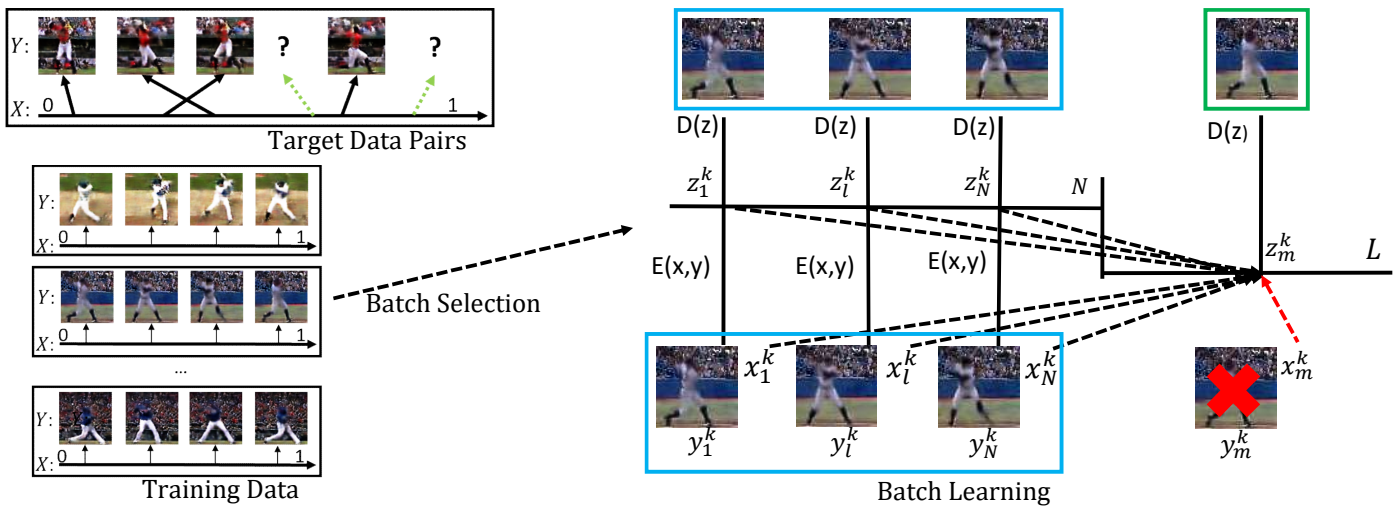


Figure 4.4: Training strategy of the proposed method. The mini-batch is generated from the sampled training data sequences.

Once the parameters are trained by the training dataset consisting of diverse golf swings, the proposed method can complete the target image sequence via regression from the incomplete test sequence on a golf swing. After training the model with the mini-batch, we fine-tune the parameters with observed data pairs in target regression.

Mini-Batch Training: The work in [96] reports that the composition of the mini-batch is critical when using variants of stochastic gradient descent methods [70, 97] to train the parameters. To generate the batch, in this paper, K sequences of a total V sequences are randomly selected. For each selected training sequence $k = 1 \cdots K$, we randomly pick L data pairs $(x_l^k, y_l^k), l = 1, ..., L$, where $L = (M + N)$. For the earlier N data pairs $(x_n^k, y_n^k), n = 1, \cdots, N$, we get the latent space vector z_n^k from the encoder function $E(y_n^k; W_E)$, and $f(x_n^k; W_x)$ to train W_E, W_x, and W_D. Alternatively, for the latter M data pairs $(x_m^k, y_m^k), m = (N + 1), \cdots, L$, we obtain the latent z_m^k by regression (Section 4.3) from $\{z_1^k, \cdots, z_N^k\}$, $\{x_1^k, \cdots, x_N^k\}$ and x_m^k. The responses $\{y_{N+1}^k, \cdots, y_L^k\}$ are assumed to be unknown in the encoding process. This data set is used to train the decoder network $D(z; W_D)$ to reconstruct the proper responses not only for the z_n^k from the data pair (x_n^k, y_n^k), but also for the z_m^k which are obtained from the regression. The corresponding loss from the estimated $\hat{y}_m^k$ and the actual y_m^k refers to the the third term in (4.1). We note that it is possible to calculate the loss term because y_m^k can be used as ground truth regression response. After constructing the batch, the stochastic gradient [70] for the batch is calculated to train all parameters.

Parameter Fine-Tuning: After training the parameters W_E, W_x and W_D using the batch from the training dataset, we further fine-tune the parameters with the observed data pairs $(x_i, y_i), i = 1, \cdots, N$ in the same way as previous regression techniques [6, 15].

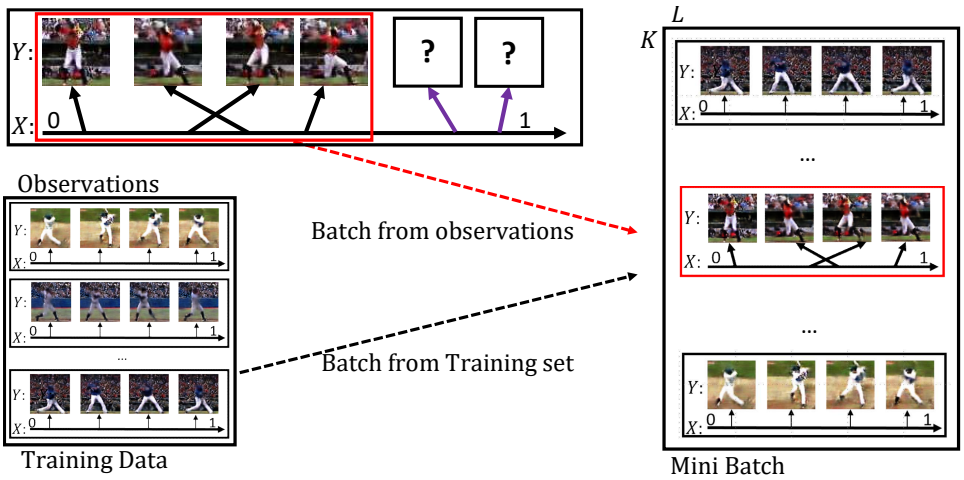


Figure 4.5: Batch generation for fine-tuning. The batch is composed of observed data pairs (red) and sampled data pairs in training dataset.

Note that the training of the regression part is not done because the ground truth is not available for the test dataset. For the fine-tuning process, mini-batches are composed of the observed test data pairs (x_i, y_i) and randomly selected $(K-1)$ data sequences (x_i^k, y_i^k) from the training set as in Fig. 4.5, where $i = 1, \cdots, N_k$ and $k = 1, ..., (K-1)$. When the total number N of observed test data pairs is less than L, we increase the number of samples by allowing repetition. Then, the parameters are fine-tuned with 50 iterations. The detailed implementation is described in the supplementary material.

4.5 Implementation Detail

In the experiments, the encoder $[m_{i,y}, \sigma_{i,y}] = E(y_i; W_E)$, decoder $D(z; W_D)$ and $\sigma(y; W_E)$ of the kernel for GP regression in Figure 2 of the paper are defined as multi-layered perceptrons. The encoder $E(y; W_E)$ is designed with five convolution layers and one fully connected layer (the convolution layers are composed of $16, 32, 64, 128,$ 256 channels with filter size 5×5 each). All $y \in \mathcal{Y}$ are resized to three channel 64-by-64 images. We set the dimension of the $m_{i,y}$ and $\sigma_{i,y}$ to 128, and the fully connected layer returns the 256 elements for $m_{i,y}$ and $\sigma_{i,y}$. The former 128 elements are used as $m_{i,y}$, and the latter 128 entries are defined as $\sigma_{i,y}$. For the mapping function $[m_{i,x}, \sigma_{i,x}] = f(x_i, W_x)$, $m_{i,x}$ refers to $x_i \in \mathcal{R}^{n(\mathcal{X})}$ and the additional $n(\mathcal{X})$ outputs in $E(y; W_E)$ indicates $\sigma_{i,x}$, as in Figure 2 of the paper. Therefore, the overall dimension of the final fully connected layer is $256 + n(\mathcal{X}) + 1$; 256 dimensions for $[m_{i,y}, \sigma_{i,y}]$, $n(\mathcal{X})$ dimensions for $\sigma_{i,x}$, and one dimension for σ_k. For the decoding function $\hat{y} = D(z; W_D)$, 6 convolution layers with 2-by-2 upsampling are used to reconstruct the image. The convolution layers have 256, 128, 64, 32, 16, 3 channels with filter size $4 \times 4, 5 \times 5, 5 \times 5, 5 \times 5, 5 \times 5$ and 5×5.

Chapter 5

Experiments

5.1 Visual Prediction

To validate the proposed algorithm, we compared the performance against the existing path prediction algorithms [33, 36]. Through the comparison, we have confirmed that the existing path prediction algorithms [33,36] are not adequate for the crowded scenes which have a temporal pattern co-occurrence tendency. Also, to check our method's applicability to pedestrian moving patterns, we compared it with Yi's method [98]. In addition, to check the effects of the components of the proposed HTGMM model, we conducted extensive experiments to evaluate our algorithm by self-comparing its performance with that of three baseline algorithms designed by naive combinations of the existing topic and Gaussian models.

5.1.1 Dataset

For the experiments, we first used QMUL [20, 100], including cross road scenes and our own complex intersection (CI) dataset captured in a wide-intersection. These scenes

Figure 5.1: Example of Crowd scene dataset, The scene in left is captured from QMUL dataset [20]. The scene in middle is captured from SNU dataset [99]. The scene in right is captured from the Grand central station, New york [98].

include diverse moving object patterns and co-occurrence types governed by traffic signals. Furthermore, these scenes are very crowded, and it is hard to utilize semantic scene segmentation information as in the previous works [33, 36]. In addition, for the pedestrian data set, we adopted PWPD [98] which does not have explicit temporal groups among the movement patterns. The dataset captures a crowded indoor plaza scene in a subway station, and the movement of the objects is far less ordered compared to the QMUL, CI datasets.

5.1.2 Comparison Methods

First, we compared the prediction performance with two major existing path prediction algorithms [33, 36] for the QMUL, CI datasets. Walker's method [36] learns the transition probability among representative mid-level patches and predicts the shape and future position of the patch. Kitani's method [36] trains the reward function for each location given semantic segmentation results and finds the predictive path which minimizes the cost. For comparison, we measured the error between ground truth trajectory and predicted trajectories of each algorithms using modified Hausdorff distance (denoted by MHD in tables) [101] and Euclidean distance (denoted by ECD in

tables). Since Walker's method automatically determines the patches for prediction, we generated ground truth trajectories for the selected patches. For the PWPD dataset, we compared the performance with Yi's method [98] which marks the state-of-the-art performance to the PWPD dataset. This method does not explicitly focus on predicting trajectories but can predict the possible destination region of objects in the scene by seeing half of the entire paths.

Second, in addition to the existing algorithms [33, 36, 98], we employed our own three baseline algorithms. In the first baseline algorithm, utilizing the movement patterns $\{\phi_k, q_k\}$ and $\theta^{(d)}$ obtained by the HTGMM, we simply inferred the co-occurrence of the movement patterns by clustering $\theta^{(d)}$ with a Gaussian mixture model. This baseline algorithm refers to 'B(1)'. The method naively breaks the proposed HTGMM into two independent models and infers the hidden variables in a greedy manner. The second baseline algorithm is designed with the same concept as the first baseline algorithm except for using $\bar{\theta}^{(d)}$ instead of $\theta^{(d)}$. The purpose of the second baseline is to show that only the simple mapping $\bar{\theta}^{(d)} = f(\theta^{(d)})$ does not give significant improvement of performance without the prior design as in the proposed HTGMM. The second baseline algorithm refers to 'B(2)'. The other baseline algorithm 'B(3)' assumes just one group. This means that the third baseline algorithm does not consider co-occurrence information. In addition, we added the prediction result obtained by humans to evaluate the prediction performance relative to human ability. Five human participants saw the training video three times repeatedly to learn the movement dynamics. They then predicted the future path from the same points given in the experiments for the proposed algorithm.

5.1.3 Qualitative Evaluation

To evaluate the robustness of design parameters, we tested our work with different parameters, namely the number of patterns K and the number of groups M. As seen

in the left graph in Figure 5.6, our method is robust in relation to K unless the number is too small. M is a more sensitive parameter than K. In the traffic scenario, we observed that selecting three to five groups achieves the best performance. It is noticeable that the performance gap is less severe if we choose a value larger than the fitted parameters. Figure 5.2 shows the patterns and their co-occurrence types extracted by the proposed algorithm. Each pattern is illustrated by utilizing regional probability ϕ_k and the potential energy map $F^{(k)}$. The co-occurrence groups of the patterns are illustrated in the right four images in each row of Figure 5.2. By utilizing the results, we measured the pattern-trajectory matching accuracy, indicating whether a trajectory is matched to an appropriate pattern in the situation at the prediction time.

CI Dataset: We set the number of patterns, K, and the number of groups, M, to 15 patterns and three co-occurrence groups, respectively, to learn the CI dataset. As shown in the right three images of Figure 5.2-(a), the proposed model can successfully make three groups with co-occurring patterns depending on the major co-occurrence types generated by traffic signals: horizontal straight, turning left with vertical straight, and vertical straight. By utilizing those patterns and groups, we conducted a prediction task and evaluated the prediction performance with 189 ground truth trajectories. As illustrated in Figure 5.3-(a), we can see that the predicted trajectories do not go toward moving objects (green arrow direction) considering co-occurrence group and arrive at the destination by following the valley obtained by the energy potential map and are matched to the ground truth. Conversely, as in Figure 5.4, the predicted path by [36] for CI data set guides cars to avoid other cars, which results in an erroneous prediction in crowded traffic conditions.

QMUL Dataset: For this dataset, we set K and M to 24 patterns and four co-occurrence groups, respectively, because the scene structure is more complicated. Figure 5.2-(b) represents the patterns and co-occurrence groups, extracted from the QMUL dataset. In Figure 5.2-(b), it is worth highlighting that the vertical straight patterns de-

Video	QMUL			CI		
Measure	precision	MHD	ECD	precision	MHD	ECD
Human	99.37	23.32	45.71	99.20	21.22	40.15
Proposed	**92.14**	**23.38**	**50.19**	**91.49%**	**27.89**	**44.95**
Proposed(2)	**-**	**11.65**	**36.80**	**-**	**14.72**	**28.60**
W14(1)	-	49.36	85.5	-	62.03	92.50
W14(2)	-	76.20	115.29	-	115.51	150.43
K12	-	86.73	107.43	-	127.62	143.60
B(1)	67.36%	41.90	71.47	63.29%	45.04	63.29
B(2)	73.14%	35.34	59.51	65.42%	43.91	56.15
B(3)	49.58%	65.70	88.05	55.31%	49.68	68.59

Table 5.1: Quantitative results of cross-street dataset. MHD indicates modified Hausdorff distance [101] and ECD denotes Euclidean distance. W14 refers to Walker's method [36] and K12 refers to Kitani's method [33]. B1,B2 and B3 indicates Baseline algorithms in section 5.1.2. W14(1) is mean value of the top 10% lowest error. W14(2) represents error of the path which has the highest probability. The result K12 is from the same configuration as W14(2).

picted by red circles in the first two groups are included in different co-occurrence groups even though they are passing the same region. Hence, their future paths will be different from each other depending on the movements of other objects. In other words, the object in the first pattern will keep going according to the vertical straight pattern, but the object in the second pattern will stop near the crosswalk region to avoid a collision with the horizontal movements. Figure 5.3-(b) shows the prediction results given the groups. We executed the prediction experiment and evaluated the performance with 246 ground truth trajectories. In this scene, there are many locations too

complicated for choosing the pattern, but our algorithm successfully selects adequate patterns for prediction. For example, the trajectory in the first image and int the second image in Figure 5.3-(b) start from almost the same location, but the predicted path is completely different depending on the co-occurrence group that is dominant at the prediction time.

PWPD Dataset: We tested our method in the pedestrian walking path dataset [98] which captures complex dynamic crowd movements. The experimental results in the PWPD dataset [98] shows that the applicability of the proposed algorithm is not restricted to cross-road traffic scenes, but can be used for a more disordered situation. Since this scene does not include the temporal group, such as traffic controlled by traffic signals, we set the number of group M to 1 and the number of patterns K to 40 via experiments which were not sensitive. We used the object trajectories given by the author [98]. As shown in Figure 5.5-(a), our model successfully learned movement patterns. Figure 5.5-(b) describes examples of path prediction results. The results show that our model successfully predicts the future when the object(human) does not loiter, as in Figure 5.5-(b).

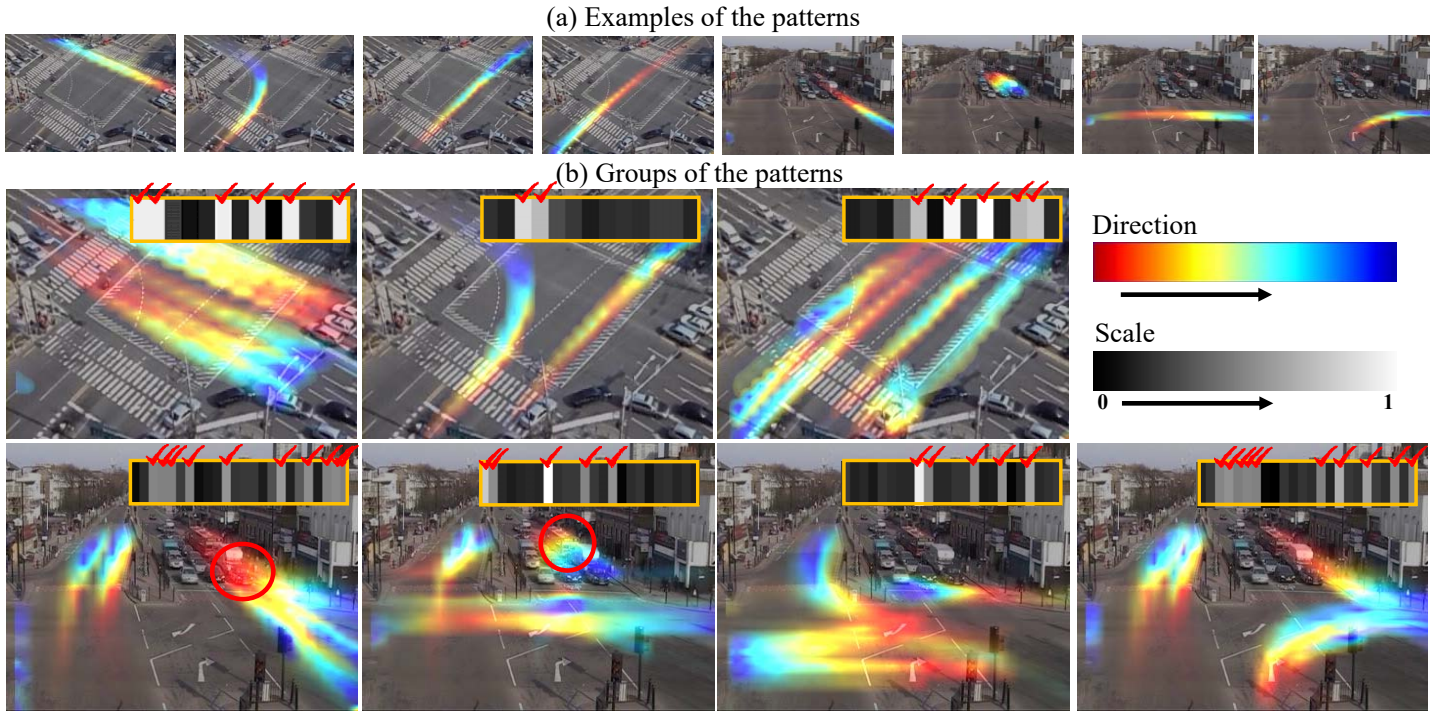


Figure 5.2: The inferred movement patterns and their co-occurrence groups. The inferred movement patterns and their co-occurrence groups. In each rows, three images in the left indicates the examples of movement patterns. Other images in the rightside depict their groups. The color of each pattern indicates the direction of the pattern, from red to blue. The bar in each picture in the rightside of each rows represents the μ_m of each Gaussian group. The gray-scaled color in the bar indicates the occurrence probability of a pattern, where a white color shows a high probability. It means that the white entries of the bar show the major patterns of the group in the corresponding picture. Best viewed in color.

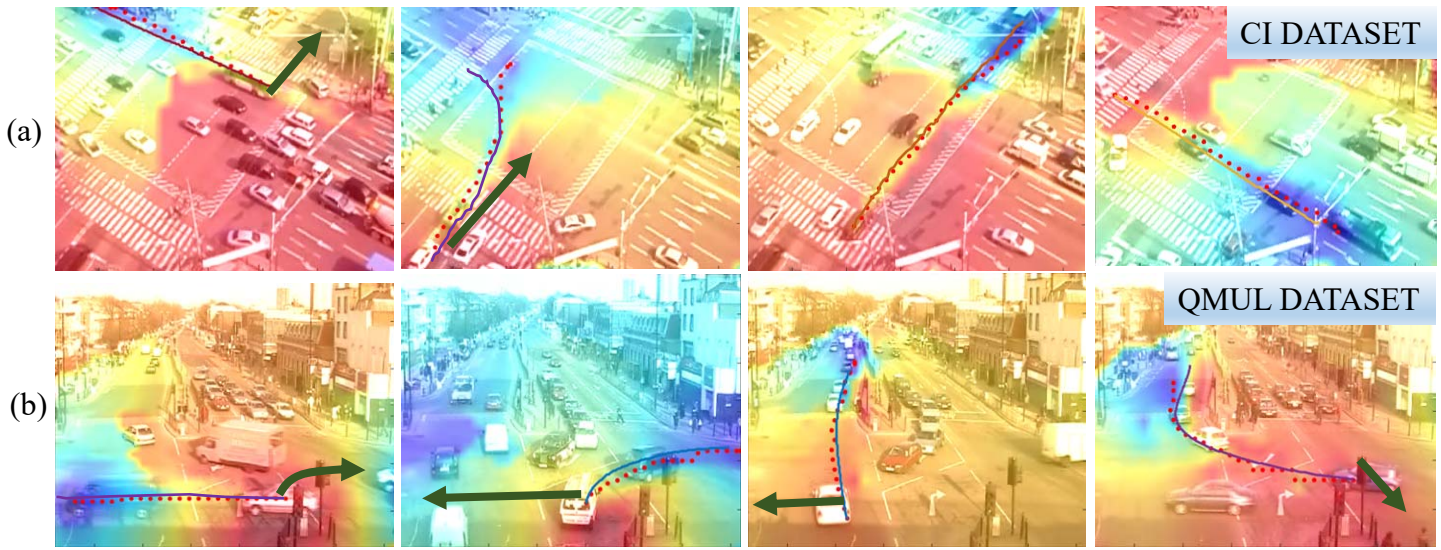


Figure 5.3: Illustration of diverse path prediction results in different groups. The solid lines indicate the ground truth trajectories and dot lines denote the predicted paths. The green arrows indicate the other possible directions if the co-occurrence groups are changed.

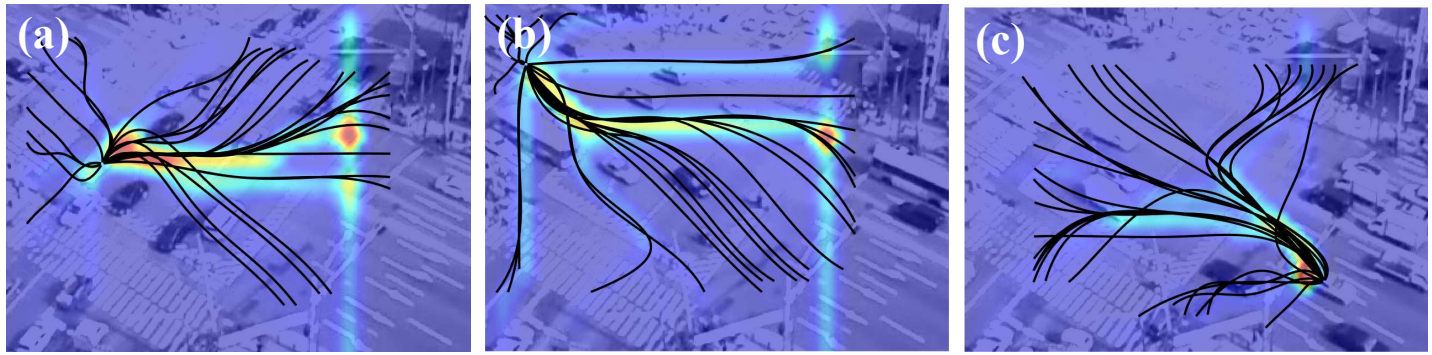


Figure 5.4: Existing Path prediction results. Path prediction results of Walkers' [36] for CI dataset.

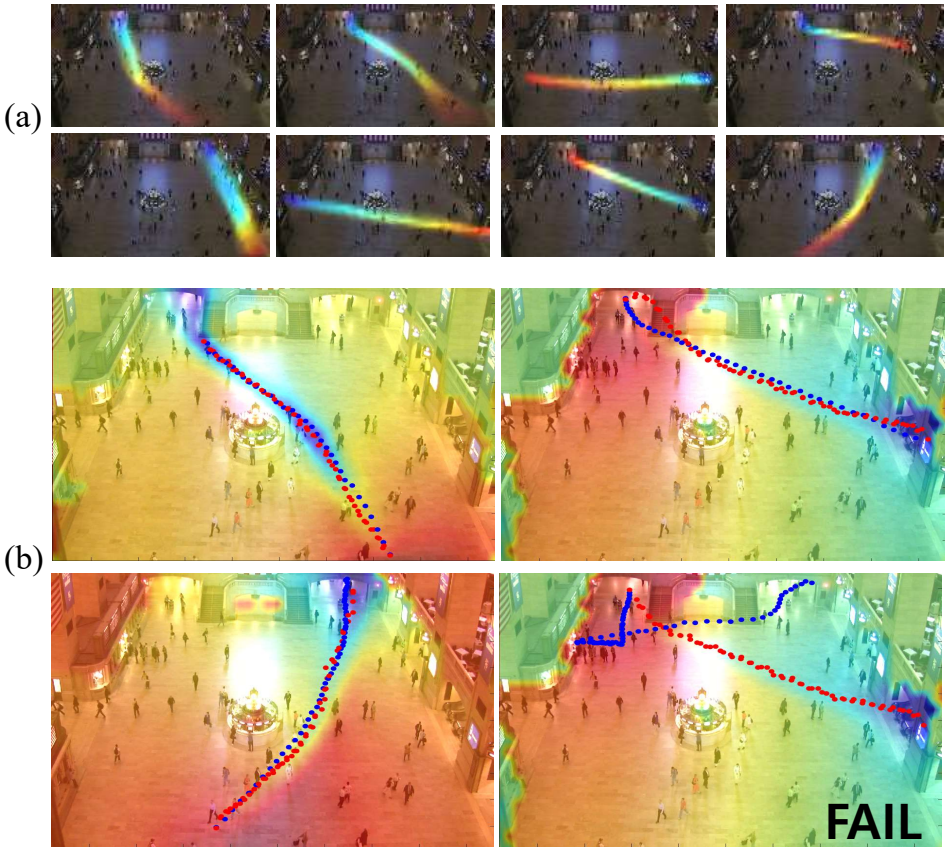


Figure 5.5: Qualitative Prediction Results for PWPD dataset [98]. (a) Extracted patterns of ours, (b) Our prediction results.

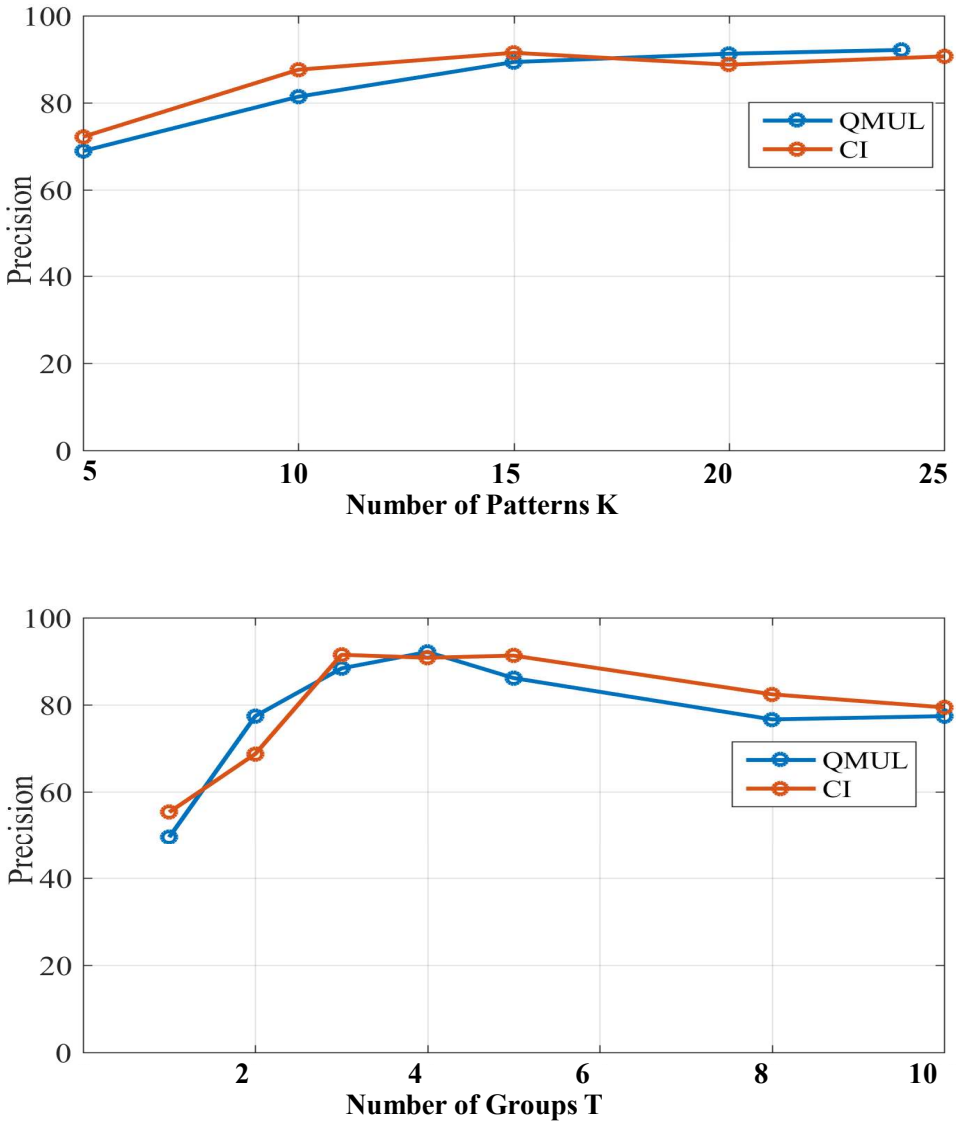


Figure 5.6: Precision Graph with respect to number of patterns and groups. Precision Graph with respect to number of patterns K and number of groups M.

5.1.4 Quantitative Evaluation

First, we conducted a quantitative comparison of the algorithms [33, 36] with the videos proposed in the paper. Table 1 shows the comparison results. For Walker's method [36], we used the mid-level features trained by the car chase dataset and the CI, QMUL datasets. For Kitani's work [33], manual ground truth segmentation results were adopted. Although the conditions of the experiment were advantageous to them, our method yielded superior performance because the two methods are designed to avoid obstacles such as cars and lawns.

We also measured the performance of the algorithm and compared the results with the baseline methods as well as with human prediction. As shown in Table 5.1, the proposed algorithm outperformed the other baseline algorithms in both datasets. The result implies that the proposed method has a meaningful contribution compared to the naive use of the existing topic and Gaussian mixture models. The baseline algorithm 'B(1)' achieved better performance than the baseline algorithm 'B(3)' which does not group the patterns. However, since the group information learned by the first baseline algorithm was inaccurate, the performance improvement by GMM was insufficient. Considering that the baseline algorithm uses the same $\theta^{(d)}$ learned by the proposed HTGMM, we conclude that the performance jump of the proposed method in comparison with the baseline algorithm 'B(1)' validates our model's superior ability to group co-occurring patterns. The result of 'B(2)' implies that utilizing sigmoid function without the proposed conjugate prior design in HTGMM does not yield a good performance. Furthermore, the prediction result (MHD, ECD) from humans and 'Proposed(2)' shows that our algorithm has a comparable prediction ability to that of humans in view of distance error. The result of 'Human' in Table 5.1 is the average value for the five humans. Interestingly, even humans were confused in predicting the path of the target, which can go in multiple directions depending on the co-occurrence

Video	measure	Y15(1)	Y15(2)	Y15(3)	Ours
PWPD	Precision	48%	38%	33%	**43.2**%

Table 5.2: Pedestrian destination results. Y15(1) [98] is the result which uses the stationary crowd factor. Y15(2),(3) are the baselines which do not, or naively use the factor.

dynamics.

Also, as shown in Table 5.2, our method achieved destination predicting performance comparable to the newest method [98] without employing the stationary crowd information, claimed to be the essential feature by Yi *et. al* [98] for analyzing a crowded scene like the PWPD dataset. It is noted that our method outperforms the other baselines of [98]: Y15(2), Y15(3), which do not utilize that factor.

5.1.5 Summary

In this chapter, we have proposed a novel path prediction algorithm that considers the moving dynamics of co-occurring objects. To solve the problem, we first designed two-layered probabilistic model to extract the major movement patterns and their co-occurrence groups in a scene. Utilizing the result from the proposed model, we have presented an effective path prediction method. By extensive qualitative/quantitative experiments, we have shown that our algorithm can predict the future paths of objects in complex scenes including many moving objects and changing situations such as cross streets with traffic lights. This paper explores a meaningful progress in path prediction research in that the proposed algorithm considers the other co-occurring objects as well as the target itself.

5.2 Visual Regression

In the experiments, we evaluated the regression capability of the proposed method via two applications composed of image data: (1) a problem with a simple temporal domain and complicated codomain and (2) a problem with a complicated domain and codomain. For the first application, we used sport data sequences obtained from YouTube. Human pose reconstruction for a given skeleton was tested for the second application.

5.2.1 Dataset

For testing sports data sequences, we created data sets for three sport sequences: baseball swing, golf swing, and weightlifting. The dataset includes 236 baseball swings, 232 golf swings and 129 weightlifting sequences from YouTube. In the dataset, $1000-2000$ images are included for each action sequence, and their relative orders are given. The domain is defined as $\mathcal{X} : [0, 1]$ and a point in $\mathcal{X}$ is assigned to x for each image $y \in \mathcal{Y}$ according to its relative order in the entire sequence. For testing, the golf and the baseball swings were trained with 200 randomly selected sequences and tested with those that remained. The weight lifting scenario was trained with 100 sequences.

For human pose reconsturction, we have used the human 3.6 million (H3.6m) [102] dataset for generating proper human appearance given the joint positions. The dataset provides 32 joint positions, and thus the input data lie in 96 dimensional space. The dataset includes diverse actions, and each action is repeatedly performed by different actors.

5.2.2 Sports Data Sequences

Evaluation Scenario:

We executed the regression for each test sequence with 20 observed images within

all images of each sequence and compared the results with multiple-output GP regression (MOGP) [9], and GP regression combined with vanilla VAE [57] (called R-VAE from here on). For R-VAE, we conducted the fine-tuning process in the same way as the proposed method. For MOGP, we trained the kernel with two-thirds of the images in the given sequences.

Qualitative Analysis: Fig. 5.7 shows the qualitative comparison of image generation results. The sequences in Fig. 5.7 show samples uniformly picked among the regressed responses from 100 evenly divided points in the range $[0, 1]$. As seen in (a), the proposed method generated the most accurate responses compared to the other methods. R-VAE also succeeded in capturing the blunt characteristics of the background and the motions of the actions. However, the generated images in (b) suffer from large amount of noise for some images it is difficult to recognize the motion (circled in red). Demonstrated are also instances in which the order of the image was not matched (circled in blue), and instances in which the background of the image was not matched (circled in green). The images in the box show the samples of reconstruction results for given image pairs. Both, the proposed method and R-VAE successfully reconstructed the images, but the regression performance is largely different. As in (c), MOGP was not successful in describing the motion changes in the image, where every regression converged to the average of the training images.

Figure 5.8, Figure 5.9 and Figure 5.10 show additional generation results of sports sequences. The figures describe the regression results of the proposed method and of R-VAE in our work. We confirmed that the proposed method achieved a superior regression performance for diverse action sequences compared to R-VAE.

Fig. 5.11 shows the effect of the fine-tuning process. The first and second column show the results with and without fine tuning. The result of the proposed method is shown in the first row and that of R-VAE in the second row. Before fine tuning, both methods generated noisy outputs, but the proposed method captured the vast character-

istics of the background as well as the change of the motions. In R-VAE, background information was less accurate than the proposed method (circled in red). After the fine-tuning process, both methods accurately reconstructed the given image pairs, as in (c). Nevertheless, the regression performance between the methods varied significantly, as in (b).

Fig. 5.12 represents the image generation results for different standard deviations. As with the original GP regression, the proposed method estimates the output responses in the form of mean and variance because the latent z for reconstructing the image is sampled from Gaussian distribution, as in (4.7). As seen in (a), the proposed algorithm captured the core semantics of the motion in each image despite the deviation change. In R-VAE, the regression results were plausible when the sampled latent z was close to the mean, but the motion in the image was regressed by a totally different action when adding large amounts of noise (up to 1.0σ). From this result, we can see that R-VAE also has an ability to align the images in the latent space according to their order as reported in previous works [57, 66]. However, the results also show that the learned variance of R-VAE does not represent the motion semantics required for regression well, which is essential for the realization of GP regression in the image space.

Fig. 5.13 shows the effect of adding domain knowledge to the latent space, as introduced in Fig. 4.2. In (a), the clusters projected in latent space are illustrated by two dimensional space using t-sne [103], which shows the images from each video sequence to be densely clustered.

The images in (b) show the effect of adding domain knowledge to the latent space. The result in the first row is from the proposed latent space, and the second row depicts the result without the additional domain knowledge. As shown in (b), the latent vector without the domain knowledge does not include sufficient semantics to provide discriminant images.

Table 5.3: Measure for the results with / without background.

Structural Similarity Index Measure [104] result			
sports	**Proposed**	**R-VAE [57]**	MIGP [9]
Baseball	**0.610 / 0.607**	**0.492 / 0.489**	0.8030 / 0.247
Golf	**0.752 / 0.707**	**0.578 / 0.543**	0.845 / 0.114
Snatch	**0.377 / 0.369**	**0.207 / 0.205**	0.626 / 0.019

Table 5.4: Measure for images from +0.5σ, $+1.0\sigma$ and $+1.5\sigma$.

SSIM result for different standard deviations				
sports	method	+0.5σ	+1.0σ	+1.5σ
Baseball	proposed	0.6453	0.5980	0.5307
	R-VAE [57]	0.4993	0.4402	0.3825
Golf	proposed	0.7203	0.4839	0.4422
	R-VAE	0.5642	0.4026	0.2417
Snatch	proposed	0.4042	0.3656	0.3629
	R-VAE	0.2700	0.1645	0.0770

Quantitative Analysis: The quantitative performance was measured using the Structural Similarity Index Measure (SSIM) [104] which captures the structural similarities between two images. We estimated the 100 images in the test set by using their domain information only, and compared the similarity between the ground truth image and the regression results. Table 5.3 shows the performance measures for generated regression images. For the three different sport sequences, the proposed method generated more similar images to the ground truth (GT) compared to R-VAE. Interestingly, the results of MOGP [9] which converged to the average of the images were measured to be most similar among the tested methods when including the background. This is because the background of the average image is almost the same as the background of the GT when the background of the GT is fixed. When we measured the similarity without the background region, MOGP was not successful and the proposed algorithm achieved the highest performance. Table 5.4 and Fig. 5.12 show the performance when changing the standard deviation. We confirmed that the proposed method generated more plausible output than R-VAE for all cases.

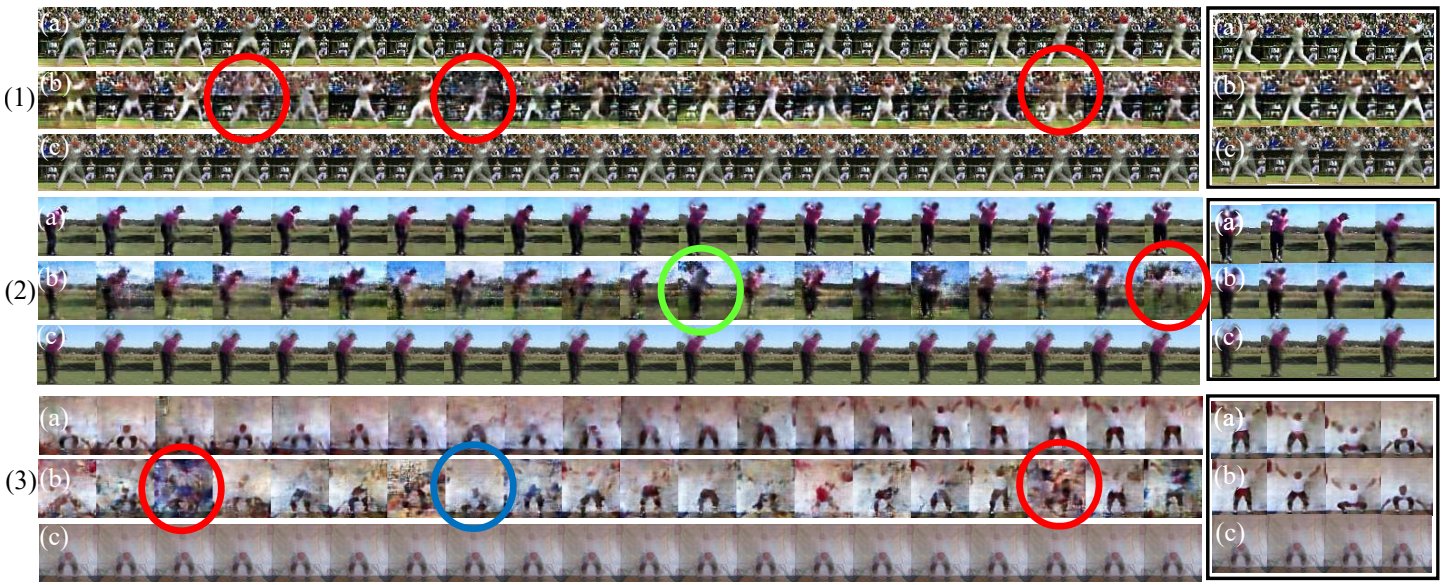


Figure 5.7: Qualitative Results on regression from the sport dataset. The row (a) in each sport represents the proposed regression results. The images in rows (b) result from the regression with R-VAE. Row (c) is the result from MOGP [9]. The results on the right indicate the samples of reconstruction results for observed images (best viewed in color).

Figure 5.8: Qualitative results on regression from the baseball swing dataset. The first row in each action represents the proposed method and the second row shows the result from R-VAE.

Figure 5.9: Qualitative results on regression from the golf swing dataset. The first row in each action represents the proposed method and the second row shows the result from R-VAE.

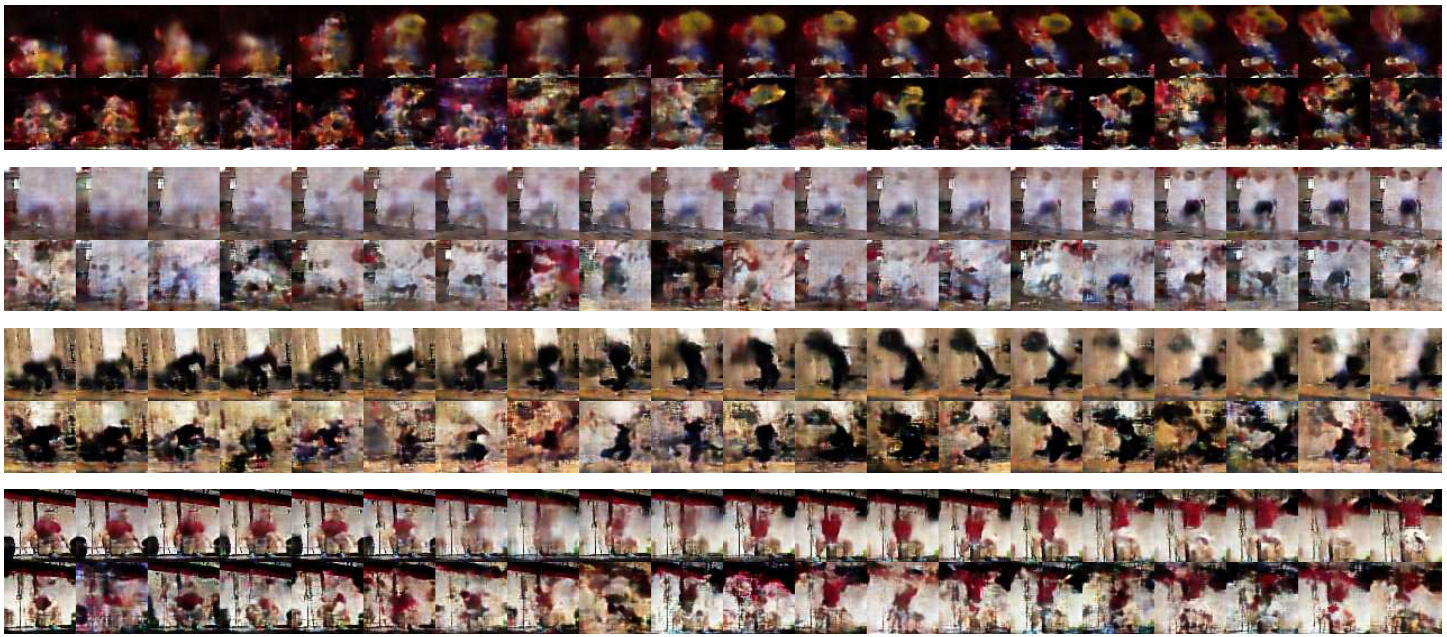


Figure 5.10: Qualitative results on regression from the weightlifting dataset. The first row in each action represents the propsoed method and the second row shows the result from R-VAE.

(a) Without fine-tuning (b) With fine-tuning (c) Reconstructed observations

Figure 5.11: Analysis on the effect of fine-tuning. (a), (b): the regression result of the proposed method is shown in the first row and that of R-VAE in the second row. (c): the images in the box denotes the samples of reconstruction results for observed images.

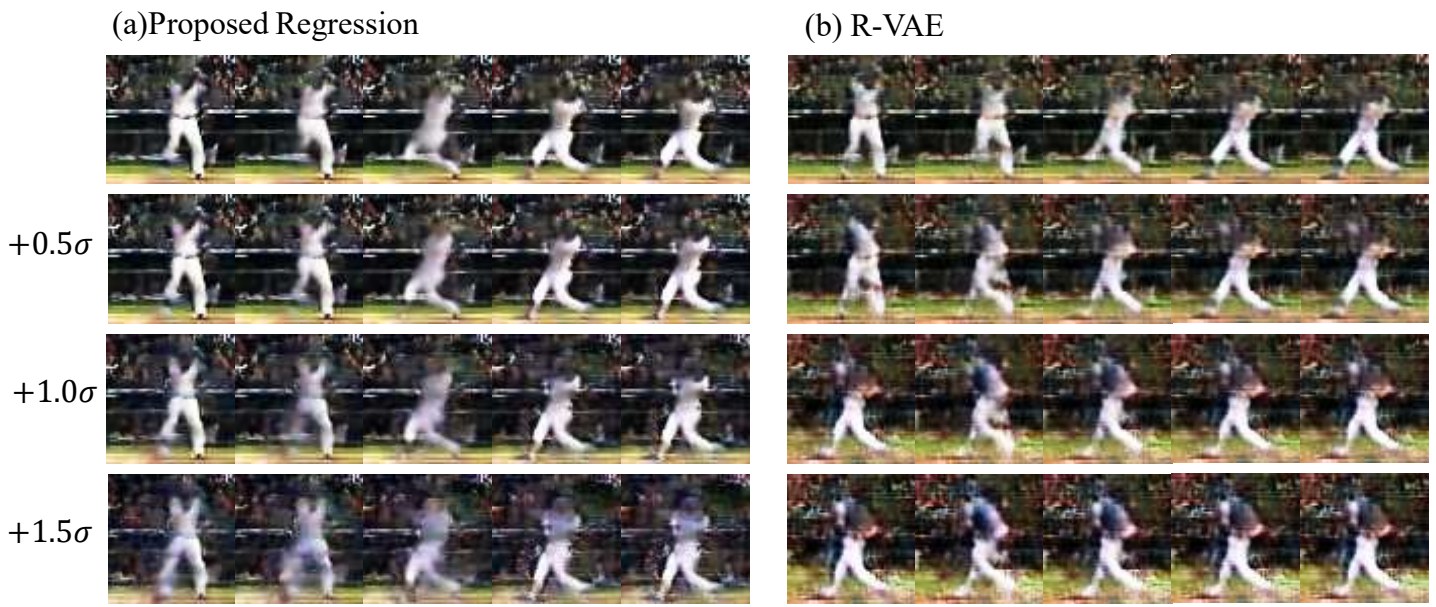


Figure 5.12: Results from +0.5σ, 1.0σ and 1.5σ latent sample.

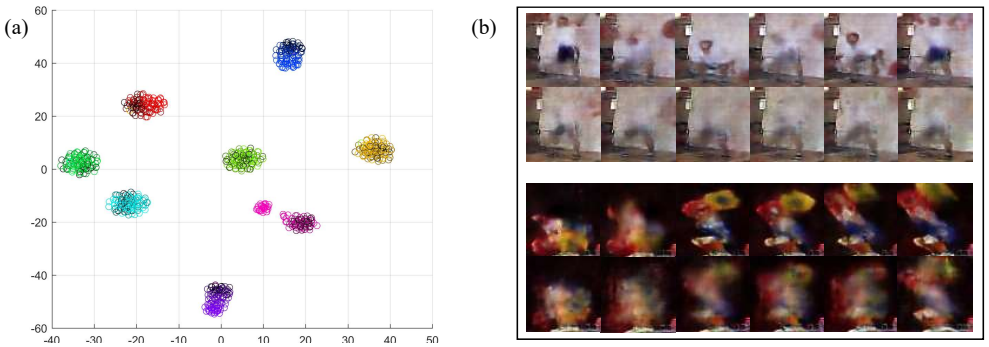


Figure 5.13: The effect of the domain knowledge in the latent space. (a) latent vector clusters when adding domain knowledge, (b) generation with (upper row) / without (lower row) domain knowledge.

5.2.3 Human Pose Reconstruction

Evaluation Scenario:

Our goal is to estimate the proper image of a new skeleton by utilizing the observed pairs of joint positions and images. In the experiment, we used the 'greeting' and 'posing' scenarios of the H3.6m dataset. The scenario for each actor was captured in 8 different view-points, resulting in a total of 16 human pose sequences available for each actor. We trained the model with the motions of 4 different actors using 12 sequences from each actor. Then, we picked the observations from the remaining four sequences and conducted the regression. The joint vectors for the regression were selected from the sequences from which the observations were selected. The joint vectors from other actors were also tested. For comparison, we used the recent conditional VAE (C-VAE) [63] method, which generated an image according to a given attribute coupled with the sampled latent code. In this experiment, the joint vector was used as the attribute.

Qualitative Analysis: Fig. 5.14 (A) shows the pose generation result of the proposed algorithm and C-VAE. For C-VAE (1), we used randomly sample latent code z_y as in [63]. For C-VAE (2), the latent code was given by the proposed regression block in Fig. 4.1. As shown in (c), the image regressed by the proposed method successfully describes the overall motion of each human pose. Also, note that the background of each image was correctly generated according to the view point of the observed data pair. The generated images from C-VAE (2) contain a large amount of noise, but they captured the rough silhouette of the actors. This result is noticeable because C-VAE usually deals with cases in which the attribute is discrete. The result from C-VAE (2) was clearer than the result from C-VAE (1), but the difference was not significant. The result in Fig. 5.14 (B) shows the output responses when the joint vectors of other actors were given. The images in the blue box refer to the ground truth pose, and the images in

the red box are the regression result by the proposed method. This result shows that the proposed method generates poses that resemble those of the input joint vectors while preserving the appearance of the given data pairs via regression. Specifically, when the given pair involves a man wearing white clothes, the generated image illustrates a man wearing the same clothes with a similar pose to the GT image. C-VAE (2) was not successful in generating a corresponding pose for a given joint from other actors.

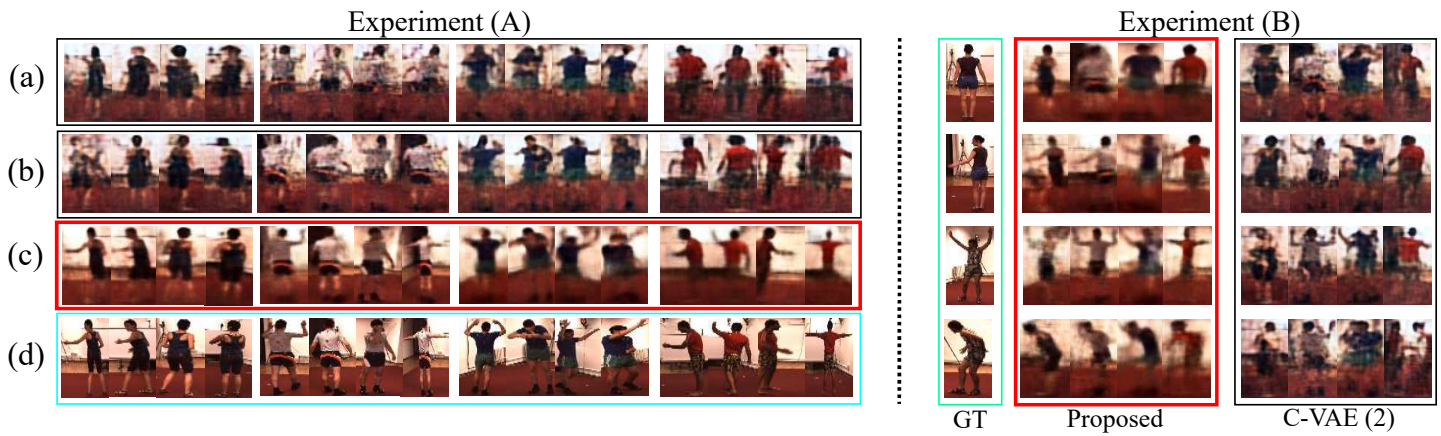


Figure 5.14: Human pose estimation results from the joint. The images in row (a) represents the C-VAE (1) results. The images in rows (b) is result from the CVAE (2). The row (c) is the result from proposed method. The row (d) shows the ground truth (best viewed in color).

Table 5.5: Similarity measure for generated human pose image.

SSIM result for human pose generation				
Actors	proposed(A)	CVAE(A)	proposed(B)	CVAE(B)
#1	0.7402	0.4849	0.5227	0.4059
#2	0.6743	0.4265	0.4775	0.3580
#3	0.7295	0.5094	0.5013	0.4268
#4	0.7671	0.4954	0.5224	0.4198

Quantitative Analysis: Table 5.5 shows the similarities between the generated image and the ground truth image. The first two columns denoted by (A) represent the quantitative results in experiment (A) of Fig. 5.14. In the experiment, the proposed method achieved a higher score than C-VAE (2). For experiment (B), we compared the similarity between the regressed image and the original images for the joint vector (green box in Fig. 5.14). There, our method also achieved a higher score than C-VAE (2).

In this experiment, the input data lay in the high dimensional space and the target joint vectors were selected without considering temporal information. Despite the complicated and non-sequential input domain, the proposed regression method achieved reasonable output responses describing the semantics given in the input and the identity information contained in the observed pairs. It means that the proposed method is available for the temporal input and can also handle more complex and non-sequential input.

Figure 5.15: Regression and reconstruction result from a same data pair. Each image set is composed of three images. Within each set, the leftmost image is generated from a regression; the middle image refers to the result from reconstruction using the joint vector and the corresponding image; and the rightmost image shows the ground truth.

To check the validity of the regression procedure conducted in a latent space, we compared the latent vector obtained by regression with the latent vector obtained by reconstruction for an input data pair. For the reconstruction, we used both the joint vector and the corresponding image in the H3.6m dataset [102]. For the regression, only the joint vector was given and the projected point was estimated by the proposed regression method. Since the latent vectors were obtained from the same input data, in ideal conditions the vectors should converge to the same location. Figure 5.15 shows the qualitative results for the regression and the reconstruction for the same input data pair, where it can be seen that both responses converged to the ground truth image. The graph in Figure 5.16 indicates the KL-divergence between the two latent vectors. Since the latent vector z in the paper is defined by a Gaussian distribution, we used the KL-divergence as distance measure. As seen in the graph, the KL-divergence obtained by the proposed method was gradually decreased. The result demonstrates that the two vectors obtained from both cases converged to the same location, as expected. When

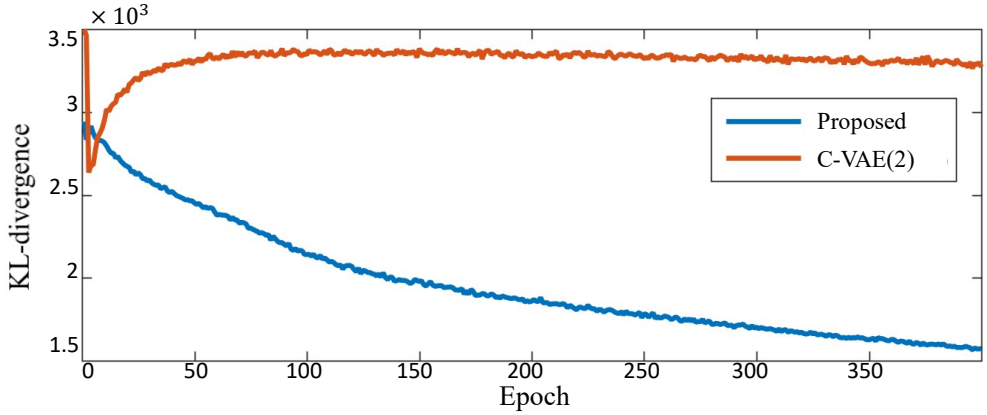


Figure 5.16: KL divergence between the latent distributions for regression and reconstruction, from same joint vectors.

tested with the C-VAE (2), the divergence did not converge.

Figure 5.8, Figure 5.9 and Figure 5.10 show additional generation results of sports sequences. The figures describe the regression results of the proposed method and of R-VAE in our work, which are the supplementary results of Figure 7 included in the submitted version. We confirmed that the proposed method achieved a superior regression performance for diverse action sequences compared to R-VAE.

5.2.4 Summary

In this chapter, we have proposed a novel regression method regarding high dimensional visual output. To tackle the challenge, the proposed regression method is designed so that the result of the regressed response in a latent space should coincide with the corresponding response in the data space. Through qualitative and quantitative analysis, it has been verified that our method properly estimates the regressed image responses and offers an approximation of the complicated input-output relationship. This paper discovers meaningful progress in the regression field in that our work

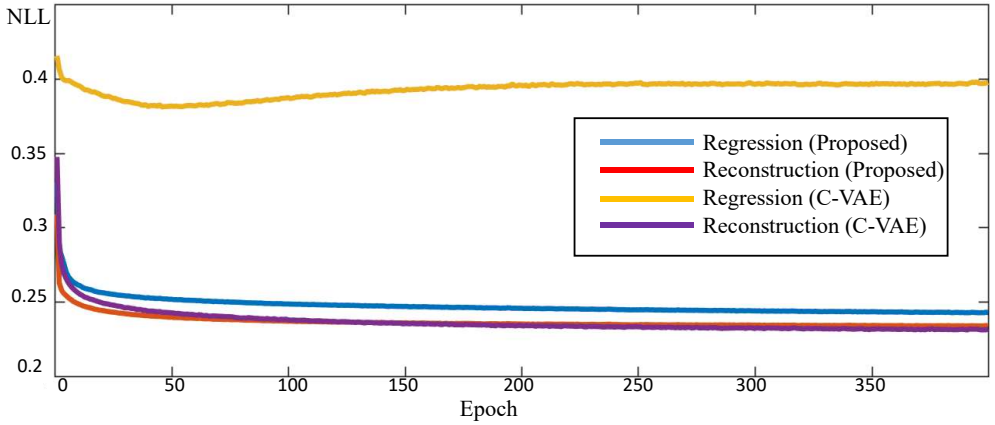


Figure 5.17: Negative Log likelihood ratio for regressed and reconstructed visual responses, from proposed method and C-VAE (2).

introduces a way to combine a deep layered architecture to the regression method in a probabilistic framework.

Chapter 6

Conclusion

6.1 Contribution

In this dissertation, we studied the prediction / regression methods for complex and high-dimensional data. To handle the data in high-dimensional space, we compressed the data into latent space embedding important semantics of the data. Then we proposed prediction / regression methods by manipulating the latent information embedded in the latent space. First, the prediction method was proposed to handle the motion dynamics latent in an image stream. Second, the regression method was introduced to estimate unobserved images from the observed images. In the prediction model, it is a meaningful contribution that an efficient sampling based inference method is proposed for the Bayesian model which combines topic mixture model and Gaussian mixture model. The combined model is difficult to be inferred by the existing sampling methods because of its large solution space and unmatched conjugate prior. We proposed a breakthrough to the problem by employing the augmented variables for the process. In addition, the extracted movement semantics from the proposed model are

mapped to continuous potential space, which is advantageous for path prediction. In the regression method, a new regression method combining the deep layered network and Gaussian process regression was firstly proposed. In the method, a method for training encoder and decoder composed of the deep layered network was proposed to let the regression in the latent space coincide with regression in the data space. The whole process is designed as a variational autoencoded framework and, to the best of our knowledge, it is the first attempt for using variational autoencoder framework to solve the regression problem. The method greatly reduced the dimension of the output responses and it is greatly beneficial for efficient regression of the data with high-dimensional responses.

6.2 Future work

In the prediction problem, we used the extracted trajectory information to predict the future position of an object at an arbitrary position in the image. The extracted trajectory information plays a key role for the crowd scene dataset used, but it is difficult to use the information in the video where the viewpoint changes. Therefore, in future research, it is interesting to develop an algorithm that can predict the image at the future time using the image information itself. Also, in the current algorithm, the training should be performed using the same video as the test video. In the future, however, it is planned to design an algorithm that learns information necessary for visual prediction using various training videos. In regression model, due to the limitations of current VAE technology, we performed regression using small size images. In the future, it can be a good research topic to pursue how to increase the image size that can be processed. Also, we also plan to apply the proposed algorithm to various applications such as data compression and large image processing. Finally, it is interesting to design a system that can integrate the proposed prediction and regression.

Bibliography

[1] He He and Wan-Chi Siu, "Single image super-resolution using gaussian process regression," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 449–456.

[2] Haiqin Yang, Laiwan Chan, and Irwin King, "Support vector machine regression for volatile stock market prediction," in *International Conference on Intelligent Data Engineering and Automated Learning*. Springer, 2002, pp. 391–396.

[3] Andrew W Lo and A Craig MacKinlay, "Stock market prices do not follow random walks: Evidence from a simple specification test," *Review of financial studies*, vol. 1, no. 1, pp. 41–66, 1988.

[4] Takashi Kimoto, Kazuo Asakawa, Morio Yoda, and Masakazu Takeoka, "Stock market prediction system with modular neural networks," in *Neural Networks, 1990., 1990 IJCNN International Joint Conference on*. IEEE, 1990, pp. 1–6.

[5] Yuichiro Anzai, *Pattern Recognition & Machine Learning*, Elsevier, 2012.

[6] Carl Edward Rasmussen, "Gaussian processes for machine learning," 2006.

[7] Mauricio Alvarez and Neil D Lawrence, "Sparse convolved gaussian processes for multi-output regression," in *Advances in neural information processing systems*, 2009, pp. 57–64.

[8] Mauricio A Alvarez, David Luengo, Michalis K Titsias, and Neil D Lawrence, "Efficient multioutput gaussian processes through variational inducing kernels.," in *AISTATS*, 2010, vol. 9, pp. 25–32.

[9] Mauricio A Alvarez and Neil D Lawrence, "Computationally efficient convolved multiple output gaussian processes," *Journal of Machine Learning Research*, vol. 12, no. May, pp. 1459–1500, 2011.

[10] Kevin Swersky, Jasper Snoek, and Ryan P Adams, "Multi-task bayesian optimization," in *Advances in neural information processing systems*, 2013, pp. 2004–2012.

[11] Carl Edward Rasmussen, "The infinite gaussian mixture model.," in *NIPS*, 1999, vol. 12, pp. 554–560.

[12] Andrew Y Ng, Michael I Jordan, Yair Weiss, et al., "On spectral clustering: Analysis and an algorithm," *Advances in neural information processing systems*, vol. 2, pp. 849–856, 2002.

[13] Hui Zou, Trevor Hastie, and Robert Tibshirani, "Sparse principal component analysis," *Journal of computational and graphical statistics*, vol. 15, no. 2, pp. 265–286, 2006.

[14] Michael E Tipping and Christopher M Bishop, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 61, no. 3, pp. 611–622, 1999.

[15] Neil D Lawrence, "Gaussian process latent variable models for visualisation of high dimensional data," *Advances in neural information processing systems*, vol. 16, no. 3, pp. 329–336, 2004.

[16] Rudolph Emil Kalman, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[17] Hawook Jeong, Youngjoon Yoo, Kwang Moo Yi, and Jin Young Choi, "Two-stage online inference model for traffic pattern analysis and anomaly detection," *Machine vision and applications*, vol. 25, no. 6, pp. 1501–1517, 2014.

[18] Jagannadan Varadarajan, Rémi Emonet, and Jean-Marc Odobez, "A sequential topic model for mining recurrent activities from long term video logs," *International journal of computer vision*, vol. 103, no. 1, pp. 100–126, 2013.

[19] Rémi Emonet, Jagannadan Varadarajan, and Jean-Marc Odobez, "Extracting and locating temporal motifs in video scenes using a hierarchical non parametric bayesian model," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 3233–3240.

[20] Timothy Hospedales, Shaogang Gong, and Tao Xiang, "A markov clustering topic model for mining behaviour in video," in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 1165–1172.

[21] Louis Kratz and Ko Nishino, "Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 1446–1453.

[22] Daniel Kuettel, Michael D Breitenstein, Luc Van Gool, and Vittorio Ferrari, "What's going on? discovering spatio-temporal dependencies in dynamic

scenes," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 1951–1958.

[23] Xiaogang Wang, Xiaoxu Ma, and W Eric L Grimson, "Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian models," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 3, pp. 539–555, 2009.

[24] Jagannadan Varadarajan, Rémi Emonet, and Jean-Marc Odobez, "Bridging the past, present and future: Modeling scene activities from event relationships and global rules," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 2096–2103.

[25] Steven M LaValle, *Planning algorithms*, Cambridge university press, 2006.

[26] Didier Devaurs, Thierry Siméon, and Juan Cortés, "Efficient sampling-based approaches to optimal path planning in complex cost spaces," in *Algorithmic Foundations of Robotics XI*, pp. 143–159. Springer, 2015.

[27] Zhan Wei Lim, David Hsu, and Wee Sun Lee, "Adaptive informative path planning in metric spaces," in *Algorithmic Foundations of Robotics XI*, pp. 283–300. Springer, 2015.

[28] Mehdi Jalalmaab, Baris Fidan, Soo Jeon, and Paolo Falcone, "Model predictive path planning with time-varying safety constraints for highway autonomous driving," in *Advanced Robotics (ICAR), 2015 International Conference on*. IEEE, 2015, pp. 213–217.

[29] Steven M LaValle and James J Kuffner, "Randomized kinodynamic planning," *The International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001.

[30] Andrew Y Ng, Stuart J Russell, et al., "Algorithms for inverse reinforcement learning.," in *Icml*, 2000, pp. 663–670.

[31] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey, "Maximum entropy inverse reinforcement learning," in *AAAI*, 2008, pp. 1433–1438.

[32] Deepak Ramachandran and Eyal Amir, "Bayesian inverse reinforcement learning," *Urbana*, vol. 51, pp. 61801, 2007.

[33] Kris Kitani, Brian Ziebart, James Bagnell, and Martial Hebert, "Activity forecasting," *Computer Vision–ECCV 2012*, pp. 201–214, 2012.

[34] Daniel Munoz, J Andrew Bagnell, and Martial Hebert, "Stacked hierarchical labeling," in *Computer Vision–ECCV 2010*, pp. 57–70. Springer, 2010.

[35] Richard Bellman, "A markovian decision process," Tech. Rep., DTIC Document, 1957.

[36] Julian Walker, Arpan Gupta, and Martial Hebert, "Patch to the future: Unsupervised visual prediction," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, 2014, pp. 3302–3309.

[37] Carl Doersch, Saurabh Singh, Abhinav Gupta, Josef Sivic, and Alexei Efros, "What makes paris look like paris?," *ACM Transactions on Graphics*, vol. 31, no. 4, 2012.

[38] Carl Doersch, Abhinav Gupta, and Alexei A Efros, "Mid-level visual element discovery as discriminative mode seeking," in *Advances in Neural Information Processing Systems*, 2013, pp. 494–502.

[39] Ian Endres, Kevin Shih, Johnston Jiaa, and Derek Hoiem, "Learning collections of part models for object recognition," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 939–946.

[40] Mamta Juneja, Andrea Vedaldi, CV Jawahar, and Andrew Zisserman, "Blocks that shout: Distinctive parts for scene classification," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 923–930.

[41] Saurabh Singh, Abhinav Gupta, and Alexei Efros, "Unsupervised discovery of mid-level discriminative patches," *Computer Vision–ECCV 2012*, pp. 73–86, 2012.

[42] Edsger W Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[43] David M Blei, Andrew Y Ng, and Michael I Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[44] Bo-Cheng Wei, *Exponential family nonlinear models*, vol. 130, Springer Verlag, 1998.

[45] Paul W Holland and Samuel Leinhardt, "An exponential family of probability distributions for directed graphs," *Journal of the american Statistical association*, vol. 76, no. 373, pp. 33–50, 1981.

[46] Steven N MacEachern and Peter Müller, "Estimating mixture of dirichlet process models," *Journal of Computational and Graphical Statistics*, vol. 7, no. 2, pp. 223–238, 1998.

[47] Yee Whye Teh, Michael I Jordan, Matthew J Beal, and David M Blei, "Hierarchical dirichlet processes," *Journal of the american statistical association*, 2012.

[48] Leonard E Baum and Ted Petrie, "Statistical inference for probabilistic functions of finite state markov chains," *The annals of mathematical statistics*, vol. 37, no. 6, pp. 1554–1563, 1966.

[49] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[50] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015.

[51] Hyeonseob Nam and Bohyung Han, "Learning multi-domain convolutional neural networks for visual tracking," *arXiv preprint arXiv:1510.07945*, 2015.

[52] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680.

[53] Yujia Li, Kevin Swersky, and Richard Zemel, "Generative moment matching networks," in *International Conference on Machine Learning*, 2015, pp. 1718–1727.

[54] Geoffrey E Hinton, "Training products of experts by minimizing contrastive divergence," *Neural computation*, vol. 14, no. 8, pp. 1771–1800, 2002.

[55] Ruslan Salakhutdinov and Geoffrey E Hinton, "Deep boltzmann machines.," in *AISTATS*, 2009, vol. 1, p. 3.

[56] Ruslan Salakhutdinov, *Learning deep generative models*, Ph.D. thesis, University of Toronto, 2009.

[57] Diederik P Kingma and Max Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[58] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, and Ian Goodfellow, "Adversarial autoencoders," *arXiv preprint arXiv:1511.05644*, 2015.

[59] Mathieu Germain, Karol Gregor, Iain Murray, and Hugo Larochelle, "Made: masked autoencoder for distribution estimation," in *International Conference on Machine Learning*, 2015, pp. 881–889.

[60] Emily L Denton, Soumith Chintala, Rob Fergus, et al., "Deep generative image models using a laplacian pyramid of adversarial networks," in *Advances in neural information processing systems*, 2015, pp. 1486–1494.

[61] Alec Radford, Luke Metz, and Soumith Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[62] Yunchen Pu, Zhe Gan, Ricardo Henao, Xin Yuan, Chunyuan Li, Andrew Stevens, and Lawrence Carin, "Variational autoencoder for deep learning of images, labels and captions," *arXiv preprint arXiv:1609.08976*, 2016.

[63] Xinchen Yan, Jimei Yang, Kihyuk Sohn, and Honglak Lee, "Attribute2image: Conditional image generation from visual attributes," *arXiv preprint arXiv:1512.00570*, 2015.

[64] Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra, "Draw: A recurrent neural network for image generation," *arXiv preprint arXiv:1502.04623*, 2015.

[65] Rahul G Krishnan, Uri Shalit, and David Sontag, "Deep kalman filters," *arXiv preprint arXiv:1511.05121*, 2015.

[66] Ross Goroshin, Michael F Mathieu, and Yann LeCun, "Learning to linearize under uncertainty," in *Advances in Neural Information Processing Systems*, 2015, pp. 1234–1242.

[67] David M Blei, "Probabilistic topic models," *Communications of the ACM*, vol. 55, no. 4, pp. 77–84, 2012.

[68] Jean-Michel Marin, Kerrie Mengersen, and Christian P Robert, "Bayesian modelling and inference on mixtures of distributions," *Handbook of statistics*, vol. 25, no. 16, pp. 459–507, 2005.

[69] Persi Diaconis, Donald Ylvisaker, et al., "Conjugate priors for exponential families," *The Annals of statistics*, vol. 7, no. 2, pp. 269–281, 1979.

[70] Diederik Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[71] Edwin T Jaynes, *Probability theory: The logic of science*, Cambridge university press, 2003.

[72] Kevin P Murphy, *Machine learning: a probabilistic perspective*, MIT press, 2012.

[73] Christopher M Bishop, *Pattern recognition and machine learning*, springer, 2006.

[74] Herbert Robbins and Sutton Monro, "A stochastic approximation method," *The annals of mathematical statistics*, pp. 400–407, 1951.

[75] Siddhartha Chib and Edward Greenberg, "Understanding the metropolis-hastings algorithm," *The american statistician*, vol. 49, no. 4, pp. 327–335, 1995.

[76] Claus S Jensen, Uffe Kjærulff, and Augustine Kong, "Blocking gibbs sampling in very large probabilistic expert systems," *International Journal of Human-Computer Studies*, vol. 42, no. 6, pp. 647–666, 1995.

[77] Jun S Liu, "The collapsed gibbs sampler in bayesian computations with applications to a gene regulation problem," *Journal of the American Statistical Association*, vol. 89, no. 427, pp. 958–966, 1994.

[78] Ian Porteous, David Newman, Alexander Ihler, Arthur Asuncion, Padhraic Smyth, and Max Welling, "Fast collapsed gibbs sampling for latent dirichlet allocation," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2008, pp. 569–577.

[79] Solomon Kullback, *Information theory and statistics*, Courier Corporation, 1997.

[80] Edwin James George Pitman, "Sufficient statistics and intrinsic accuracy," in *Mathematical Proceedings of the cambridge Philosophical society*. Cambridge Univ Press, 1936, vol. 32, pp. 567–579.

[81] Erling Bernhard Andersen, "Sufficiency and exponential families for discrete sample spaces," *Journal of the American Statistical Association*, vol. 65, no. 331, pp. 1248–1255, 1970.

[82] Bernard Osgood Koopman, "On distributions admitting a sufficient statistic," *Transactions of the American Mathematical Society*, vol. 39, no. 3, pp. 399–409, 1936.

[83] Morton Kupperman, "Probabilities of hypotheses and information-statistics in sampling from exponential-class populations," *The Annals of Mathematical Statistics*, pp. 571–575, 1958.

[84] John Paisley, David Blei, and Michael Jordan, "Variational bayesian inference with stochastic search," *arXiv preprint arXiv:1206.6430*, 2012.

[85] Tijmen Tieleman and Geoffrey Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural networks for machine learning*, vol. 4, no. 2, 2012.

[86] Mark Ebden, "Gaussian processes for regression: A quick introduction," *The Website of Robotics Research Group in Department on Engineering Science, University of Oxford*, 2008.

[87] Carlo Tomasi and Takeo Kanade, *Detection and tracking of point features*, School of Computer Science, Carnegie Mellon Univ. Pittsburgh, 1991.

[88] Robert G Bartle, *The elements of integration and Lebesgue measure*, John Wiley & Sons, 2014.

[89] Alan E Gelfand and Adrian FM Smith, "Sampling-based approaches to calculating marginal densities," *Journal of the American statistical association*, vol. 85, no. 410, pp. 398–409, 1990.

[90] Gareth O Roberts and Sujit K Sahu, "Updating schemes, correlation structure, blocking and parameterization for the gibbs sampler," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 291–317, 1997.

[91] Daisy Zhe Wang, Eirinaios Michelakis, Minos Garofalakis, and Joseph M Hellerstein, "Bayesstore: managing large, uncertain data repositories with probabilistic graphical models," *Proceedings of the VLDB Endowment*, vol. 1, no. 1, pp. 340–351, 2008.

[92] Tom Griffiths, "Gibbs sampling in the generative model of latent dirichlet allocation," 2002.

[93] Herman Kamper, "Gibbs sampling for fitting finite and infinite gaussian mixture models," 2013.

[94] Robert J Schalkoff, *Digital image processing and computer vision*, vol. 286, Wiley New York, 1989.

[95] Matthew James Beal, *Variational algorithms for approximate Bayesian inference*, University of London London, 2003.

[96] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston, "Curriculum learning," in *Proceedings of the 26th annual international conference on machine learning*. ACM, 2009, pp. 41–48.

[97] John Duchi, Elad Hazan, and Yoram Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011.

[98] Shuai Yi, Hongsheng Li, and Xiaogang Wang, "Understanding pedestrian behaviors from stationary crowd groups," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3488–3496.

[99] YoungJoon Yoo, Kimin Yun, Sangdoo Yun, JongHee Hong, Hawook Jeong, and Jin Young Choi, "Visual path prediction in complex scenes with crowded moving objects," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[100] Kai-Wen Cheng, Yie-Tarng Chen, and Wen-Hsien Fang, "Video anomaly detection and localization using hierarchical feature representation and gaussian process regression," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2909–2917.

[101] Marie-Pierre Dubuisson and Anil K Jain, "A modified hausdorff distance for object matching," in *Pattern Recognition, 1994. Vol. 1-Conference A: Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on*. IEEE, 1994, vol. 1, pp. 566–568.

[102] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu, "Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1325–1339, jul 2014.

[103] Laurens van der Maaten and Geoffrey Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[104] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

초록

본 논문은 높은 차원의 데이터에 대한 효과적인 회귀/예측 모델에 대한 연구를 소개한다. 기존의 회귀/예측 방법들과 달리, 제안하는 모델은 출력 데이터가 영상 데이터인 경우에 대해 문제를 푼다. 이러한 영상 데이터는 그 차원이 매우 높고 복잡한 위상 표면 위에 존재하기 때문에 기존의 알고리즘에서는 효과적으로 다루지 못해왔다. 이러한 데이터를 다루기 위하여 우리는 데이터의 정보를 압축하는 잠재 공간을 설정하고, 해당 공간에서 효과적으로 회귀/예측을 수행한다. 이러한 잠재 공간의 형성과 회귀/예측 알고리즘은 하나의 베이지안 모델로써 설계된다. 본 논문에서는 먼저 시간 순으로 들어오는 영상 내 움직임 정보를 이용하는 예측 모델을 제안하고, 이후 더욱 일반적인 영상을 포함한 입력 출력 데이터를 처리할 수 있는 회귀 모델을 제안한다. 전자의 모델의 경우, 가우시안 합성 모델과 토픽 합성 모델을 이용한 계층적 토픽 가우시안 합성 모델을 제안함으로써 모션 데이터의 공간적, 시간적인 특징을 찾아내였고, 이 정보들을 이용하여 영상 내 임의의 위치의 물체에 대한 예측 모델을 개발하였다. 후자의 모델의 경우 더욱 일반적인 영상 데이터를 처리하기 위해 신경망 모델이 추가되었으며, 이를 이용 더욱 복잡한 영상 데이터에 대한 회귀 분석을 시행하였다. 이 때 신경망 구조는 영상 데이터의 정보를 낮은 차원의 공간으로 함축하며 이는 베이지안 방법론에 의해 가우시안 회귀 알고리즘과 통합되었다. 각각의 알고리즘은 다양한 정성적, 정량적 평가를 통해 그 효율성을 검증하였다.

주요어: 회귀분석, 확률적 그래프 모델, 계층적 생성 모델, 근사 추론
학번: 2011-20884